



Discussion Paper

No. 2009–52

**ENHANCEMENT OF SANDWICH ALGORITHMS FOR  
APPROXIMATING HIGHER DIMENSIONAL CONVEX PARETO  
SETS**

By Gijs Rennen, Edwin R. van Dam, Dick den Hertog

June 2009

ISSN 0924-7815

# Enhancement of sandwich algorithms for approximating higher dimensional convex Pareto sets

Gijs Rennen • Edwin R. van Dam • Dick den Hertog

*Department of Econometrics and Operations Research, Tilburg University,  
P.O. Box 90153, 5000 LE Tilburg, The Netherlands*

G.Rennen@uvt.nl • Edwin.vanDam@uvt.nl • D.denHertog@uvt.nl

## Abstract

In many fields, we come across problems where we want to optimize several conflicting objectives simultaneously. To find a good solution for such multi-objective optimization problems, an approximation of the Pareto set is often generated. In this paper, we consider the approximation of Pareto sets for problems with three or more convex objectives and with convex constraints. For these problems, sandwich algorithms can be used to determine an inner and outer approximation between which the Pareto set is 'sandwiched'. Using these two approximations, we can calculate an upper bound on the approximation error. This upper bound can be used to determine which parts of the approximations must be improved and to provide a quality guarantee to the decision maker.

In this paper, we extend higher dimensional sandwich algorithms in three different ways. Firstly, we introduce the new concept of adding dummy points to the inner approximation of a Pareto set. By using these dummy points, we can determine accurate inner and outer approximations more efficiently, i.e., using less time-consuming optimizations. Secondly, we introduce a new method for the calculation of an error measure which is easy to interpret. The combination of easy calculation and easy interpretation makes this measure very suitable for sandwich algorithms. Thirdly, we show how transforming certain objective functions can improve the results of sandwich algorithms and extend their applicability to certain non-convex problems. The calculation of the introduced error measure when using transformations will also be discussed.

To show the effect of these enhancements, we make a numerical comparison using four test cases, including a four-dimensional case from the field of intensity-modulated radiation therapy (IMRT). The results of the different cases show that we can indeed achieve an accurate approximation using significantly fewer optimizations by using the enhancements.

**Keywords:** Convexity,  $\epsilon$ -efficiency,  $\varepsilon$ -Pareto optimality, Geometric programming, Higher dimensional, Inner and outer approximation, IMRT, Pareto set, Multi-objective optimization, Sandwich algorithms, Transformations.

**JEL Classification:** C61.

## 1 Introduction

In many fields, we come across problems where we want to optimize several conflicting objectives simultaneously. An example from engineering is the optimization of ride safety and

ride comfort when designing car suspensions (Nguyen (2007)). But also in health care, optimization of multiple objectives can be an issue. When treating cancer tumors, maximizing the probability of eradicating a tumor and minimizing the probability of damaging healthy tissue are two conflicting objectives. These are just two of many possible applications of multi-objective optimization. For a comprehensive overview of the range of applications, we refer to the paper of White (1990) which lists 500 papers describing different applications in various fields.

Given the large number of applications, it is not surprising that multi-objective optimization is an active field of research. Many approaches have been developed to deal with optimizing multiple objectives. As in general there exists no solution which optimizes all objectives simultaneously, the concept of Pareto optimality is often used to find a solution for a multi-objective problem (MOP). A solution is called Pareto optimal if it is not possible to improve an objective without worsening any of the other objectives. The set of all Pareto optimal solutions is generally called the Pareto set or Pareto frontier. Many multi-objective optimization techniques are focused on approximating (part of) this Pareto set. The approximation can be used by a human decision maker to get insight into the trade-offs between the different objectives and to select a solution which best satisfies his preferences. For a general discussion of multi-objective optimization techniques, see the books of Hwang and Masud (1979), Steuer (1986), Miettinen (1999), Ehrgott (2005), and Branke et al. (2008), and the survey papers of Ruzika and Wiecek (2003), Marler and Arora (2004), and Ehrgott and Wiecek (2005).

In this paper, we consider the class of higher dimensional convex MOPs. By higher dimensional, we mean that the MOP has more than two objectives. We make this distinction because many methods only apply to two objectives. Note, however, that the methods discussed in this paper can also be applied to bi-objective convex MOPs. An MOP is called convex if all objective functions and the set of all feasible solutions are convex (Miettinen (1999)). Romeijn et al. (2004) have shown that for these problems the Pareto set is also convex. The choice for this class of problems was inspired by the intensity-modulated radiation therapy (IMRT) optimization problem. This problem deals with designing a beam fluence map which delivers enough radiation dose to a cancer tumor to eradicate it, while keeping the dose in other tissue low enough to avoid damage. Different objective functions can be used to formulate this problem as an MOP. Romeijn et al. (2004), Hoffmann et al. (2006), and Siem et al. (2008) have shown that many commonly used objective functions are convex or can be transformed into convex functions without changing the Pareto set. For an overview of papers treating the IMRT-problem as an MOP, we refer to Romeijn and Dempsey (2008) and the references therein. Note that although our research was inspired by the IMRT problem, the methods discussed and introduced in this paper apply to all convex MOPs and even certain non-convex MOPs.

There exists a wide variety of methods for approximating higher dimensional convex Pareto sets (see, e.g., Ruzika and Wiecek (2003), Marler and Arora (2004), Ehrgott and Wiecek (2005), and Karasakal and Koksalan (2009)). However, we will focus on higher dimensional sandwich algorithms because they have several interesting and useful properties. Sandwich algorithms approximate the Pareto set by iteratively improving an inner and outer approximation. As the real Pareto set is sandwiched between the inner and outer approximation, an upper bound on the approximation error can be determined using these approximations. We consider this to be a major advantage over other types of methods as they generally cannot provide information on the accuracy of the approximation. The availability of this

information has two benefits. Firstly, most sandwich algorithms use this information to determine which part of the approximation needs to be improved in each step of the algorithm. By improving the part where the upper bound on the approximation error is largest, sandwich algorithms efficiently improve the approximation. Secondly, the decision maker can use this approximation error to determine if a certain approximation is accurate enough. However, in order for the error to be useful in practice, the error measure should be easy to calculate and interpret.

As with general MOP-algorithms, many sandwich algorithms are developed for only the bi-objective cases and cannot directly be extended to higher dimensions. However, there are several sandwich algorithms that can deal with higher dimensional MOPs. Solanki et al. (1993), for instance, have extended their bi-objective method to make it suitable for higher dimensions. Although their method is described for multi-objective linear programming problems, it can also be applied to general convex MOPs. Klamroth et al. (2002) also introduced methods for generating an outer and inner approximation. The sandwich method of Craft et al. (2006) was introduced for higher dimensional IMRT-problems, but can also be applied to other convex MOPs. Lastly, also Shao and Ehrgott (2008) developed a sandwich algorithm motivated by the IMRT-problem. However, their method deals with it as a multi-objective linear programming problem and cannot easily be extended to general convex MOPs.

In this paper, we extend higher dimensional sandwich algorithms in three different ways. Firstly, we introduce the new concept of adding dummy points to the inner approximation of a Pareto set. By using these dummy points, we can determine accurate inner and outer approximations more efficiently, i.e., using less time-consuming optimizations. We illustrate this by enhancing the method of Solanki et al. (1993) with dummy points and comparing this method with existing sandwich algorithms on a number of test cases. Furthermore, certain points of the inner approximation could be irrelevant as they are dominated by other points of the inner approximation. The detection of the relevant points can also be simplified by the use of dummy points.

Secondly, we introduce an error measure which determines the quality of an approximation based on the concept of  $\epsilon$ -dominance. An important benefit of this error measure is that it provides the decision maker with quality guarantees that are easy to interpret. However, the calculation of this error measure is not straightforward. Therefore, we introduce a new calculation method which simplifies the calculations by using dummy points. When calculating an upper bound for this measure using the inner and outer approximation, the method simplifies the calculations to solving a number of relatively simple LP-problems. As the measure thus becomes easy to calculate, it can also be used in sandwich algorithms to determine which part of the approximation should be improved in each iteration. By doing this, we are likely to obtain an accurate approximation more efficiently. Furthermore, it enables the decision maker to easily evaluate the accuracy of the approximations at each iteration of the algorithm. The combination of easy calculation and easy interpretation thus makes it a very suitable measure for sandwich algorithms.

Thirdly, we show how transforming certain objective functions can improve the results of sandwich algorithms or extend their applicability. Approximations of convex Pareto sets can be improved if we can find a strictly increasing concave transformation function such that the transformed objective is still convex. We prove that both inner and outer approximations can be improved in this way. The improved accuracy means that even less time-consuming optimizations will be needed to achieve a certain accuracy. By using transformation functions, the application of sandwich algorithms can also be extended to certain non-convex MOPS. We

will show that if we can find a strictly increasing transformation function which transforms the non-convex objective functions into convex ones, we can use the sandwich algorithms to determine inner and outer approximations for non-convex Pareto sets. The calculation of the introduced error measure when using transformations will also be discussed. For bi-objective MOPs, Siem et al. (2008) already showed similar results by using transformations. Our results thus extend their findings to higher dimensional MOPs.

To show the effect of these enhancements, we make a numerical comparison using four test cases. The set of cases consists of a three-dimensional strictly convex MOP, a five-dimensional linear MOP, a four-dimensional IMRT case and a three-dimensional geometric programming case. The results of the different cases show that we indeed need significantly fewer optimizations to achieve an accurate approximation when using the enhancements.

This paper is organized as follows. In Section 2, we give a formal definition of the convex MOP and introduce the necessary notation. Section 3 contains descriptions of the above mentioned higher dimensional sandwich algorithms. The concept of dummy points and their application is introduced in Section 4. In Section 5, we motivate and define the previously mentioned error measure and show how we can easily calculate it by using dummy points. The transformation of objective functions is discussed in Section 6. Calculation of the error measure when using transformations is also discussed in this section. In Section 7, we describe how the dummy points, error measure and transformation functions can be applied in several sandwich and non-sandwich algorithms for approximating MOPs. To compare the different sandwich algorithms and to show the improvements from the above enhancements, Section 8 contains a numerical comparison consisting of four test cases. Finally, Section 9 finishes with concluding remarks.

## 2 Problem definition and notation

Throughout this paper, we use the following orderings of vectors. Let  $x, y \in \mathbb{R}^n$  with  $n \geq 2$ . With  $x_i$ , we denote the  $i^{th}$  element of the vector  $x$ . To enumerate different vectors, we use superscripts. When ordering two vectors, we use:

- $x < y \Leftrightarrow x_i < y_i$  for all  $i = 1, \dots, n$ .
- $x \leq y \Leftrightarrow x_i \leq y_i$  for all  $i = 1, \dots, n$  and  $x \neq y$ .
- $x \leq y \Leftrightarrow x_i \leq y_i$  for all  $i = 1, \dots, n$ .

The symbols  $>$ ,  $\geq$ ,  $\leq$  are defined accordingly. We furthermore define the set  $\mathbb{R}_{\leq}^n = \{x \in \mathbb{R}^n : x \leq 0\}$ . If  $X \subset \mathbb{R}^n$ , then we define  $X + \mathbb{R}_{\leq}^n = \{y \mid \exists x \in X : y \leq x\}$ . The sets  $\mathbb{R}_{\geq}^n$ ,  $\mathbb{R}_{\leq}^n$ , and  $\mathbb{R}_{\geq}^n$  and the sets  $X + \mathbb{R}_{\geq}^n$ ,  $X + \mathbb{R}_{\leq}^n$ , and  $X + \mathbb{R}_{\geq}^n$  are again defined accordingly.

In this paper, we consider the following multi-objective optimization problem (MOP):

$$\begin{aligned} \min f(x) &= [f_1(x), \dots, f_m(x)]^\top \\ \text{s.t. } x &\in X \\ f(x) &\leq z^{ub}, \end{aligned} \tag{1}$$

where  $X$  is the feasible set,  $f : X \rightarrow \mathbb{R}_{\geq}^m$  is the vector of  $m$  real valued objective functions and  $z^{ub}$  is an upper bound on the objective function values. Note that this definition differs in

two ways from the common MOP. Firstly, we assume that all objective functions give values greater than or equal to zero. This is no real limitation as adding a fixed constant to each objective value does not essentially change the problem or solution set. By simply adding the utopia point, which we define further on, we can make sure that an arbitrary objective function satisfies this condition. Secondly, the problem is usually formulated without the upper bound restriction on the objectives. However, in practice the decision maker generally has an idea of the maximal value he is willing to accept for the different objectives. Solutions with higher objective values are irrelevant and can thus be avoided using this restriction.

As it is generally impossible to find an  $x \in X$  that minimizes all objectives at the same time, our aim is to find a set of so-called Pareto optimal or non-dominated solutions.

**Definition 1.**

*An objective vector  $f(x)$  for  $x \in X$  is (strongly) dominated if there exists an  $\tilde{x} \in X$  such that  $f(\tilde{x}) < f(x)$ .*

*An objective vector  $f(x)$  for  $x \in X$  is weakly dominated if there exists an  $\tilde{x} \in X$  such that  $f(\tilde{x}) \leq f(x)$ .*

**Definition 2.**

*An objective vector  $f(x)$  for  $x \in X$  is (strongly) Pareto optimal if there exists no  $\tilde{x} \in X$  such that  $f(\tilde{x}) \leq f(x)$ .*

*An objective vector  $f(x)$  for  $x \in X$  is weakly Pareto optimal if there exists no  $\tilde{x} \in X$  such that  $f(\tilde{x}) < f(x)$ .*

Pareto optimality thus implies that it is not possible to improve one objective without deteriorating at least one other objective. If, on the other hand, this is possible, then  $f(x)$  is weakly dominated. When it is possible to improve on all objectives, then  $f(x)$  is strongly dominated. The concept of Pareto optimality is also known under the names efficiency and non-dominance. The different terms are sometimes used to distinguish between points in the design space and objective space, but as Ehrgott (2005) points out, there is no general consensus on which term should be used for which space. In this paper, we choose to use efficiency for points in the design space and Pareto optimality for points in the objective space. The set of all feasible efficient solutions is thus a subset of  $X$  and will be denoted by  $X_E$ . The set containing all the Pareto optimal vectors corresponding to the efficient solutions, i.e.,  $PS := \{f(x) \mid x \in X_E, f(x) \leq z^{ub}\}$ , is called the Pareto set. The set  $PS$  is a subset of the set  $Z$  of all feasible criterion vectors, which is defined as  $Z := \{f(x) \mid x \in X, f(x) \leq z^{ub}\}$ . We shall denote vectors in the solution space by  $x$  and vectors in the objective space by  $z$ . Furthermore, we refer to vectors  $x$  as solutions and vectors  $z$  as points.

We assume that the decision maker has selected the upper bounds  $z^{ub}$  such that all solutions in  $PS$  are viable solutions. The decision maker is thus interested in the complete Pareto set. However, as determining the complete Pareto set is generally impossible, we try to approximate it. To determine Pareto optimal points, an often used method is to solve the following weighted-sum problem:

$$z^* = \arg \min \{w^\top z \mid z \in Z\}, \quad (2)$$

where  $w \in \mathbb{R}_{\geq}^m$ . Advantages of this problem formulation are that it is often easy to implement and that the problem is convex for convex MOP. Furthermore for  $w > 0$ , solving this problem always produces a Pareto optimal point. If  $w_i = 0$  for one or more  $i \in \{1, \dots, m\}$  then the

resulting point is also guaranteed Pareto optimal if all corresponding objective functions  $f_i(x)$  are strongly convex. However, when some of the corresponding objective functions  $f_i(x)$  are weakly convex, the resulting point can be weakly Pareto optimal (see, e.g., Miettinen (1999)). To determine a point  $z^{**}$  which is guaranteed Pareto optimal and which weakly dominates or is equal to  $z^*$ , we can solve the following additional optimization problem:

$$z^{**} = \arg \min \{(w')^T z \mid z \leq z^*, z \in Z\},$$

where  $w' > 0$ . However, as we assume that the optimizations are time-consuming and the benefit of this additional optimization may be relatively small, we do not perform this step for the test-cases in this paper.

A common approach to approximate a Pareto set is to solve the weighted sum problem for an even spread of  $w$ . However as Das and Dennis (1997) have shown, this method does not generally give an even spread of points from a Pareto set. We therefore need a more advanced method to efficiently determine a good approximation of the Pareto set.

Besides general Pareto points, several auxiliary points in the objective space are often used to approximate the Pareto set. An anchor point  $z^{Ai}$  of an MOP is a point in  $Z$  defined as:

$$z^{Ai} = \arg \min \{z_i \mid z \in Z\} \text{ for } i = 1, \dots, m.$$

At an anchor point, one of the objectives is thus minimized without taking the other objectives into account. Note that anchor points can be found by solving the weighted-sum problem with  $w$  equal to the  $i^{th}$  unit vector. The results concerning Pareto optimality of solutions of the weighted-sum problem thus also apply to anchor points.

The utopia point  $z^U$  is found by taking the minimal values of all the objectives:

$$z_i^U = z_i^{Ai} \text{ for } i = 1, \dots, m.$$

The utopia point is thus the best possible point according to all objectives, but in general it is infeasible. The nadir point is the opposite of the utopia point and obtained by:

$$z_i^N = \max \{z_i \mid z \in PS\} \text{ for } i = 1, \dots, m.$$

Even though the addition of the upper bound  $z^{ub}$  to the common MOP formulation can simplify the computations, determining the nadir point for  $m > 2$  remains in general hard as we have to optimize over the unknown set  $PS$  (see, e.g., Miettinen (1999)). In Schandl et al. (2002), the following generalization of the nadir-point concept is proposed:

$$z_i^{pN} = \max \{z_i^{Aj} \mid j = 1, \dots, m\} \text{ for } i = 1, \dots, m.$$

We refer to this point as the pseudo-nadir point.

Furthermore, as stated in the introduction, we assume that the MOP is convex, i.e., all objective functions  $f_i(x)$  and the set  $X$  are convex. According to Romeijn et al. (2004), this implies that the set:

$$Z + \mathbb{R}_{\geq}^m = \{z \mid \exists \tilde{z} \in Z : z \geq \tilde{z}\},$$

of dominated points is also convex. In Jin and Sendhoff (2004), the convexity of a Pareto set is formulated as follows.

**Definition 3.** A set  $PS$  is convex if for all  $u, v \in PS$  and for all  $\lambda \in (0, 1)$ , there exists a vector  $w \in PS$  such that  $\lambda u + (1 - \lambda)v \geq w$ .

The result of Romeijn et al. (2004) implies that also convexity according to the definition of Jin and Sendhoff (2004) is satisfied for convex MOP.

To approximate a convex Pareto set, we often use the convex hull of a set of points in  $Z$ . For  $Y \subset Z$ , the convex hull  $\text{conv}\{Y\}$  is defined as the set of all convex combinations of points in  $Y$ . All points in  $\text{conv}\{Y\}$  which cannot be written as a convex combination of other points in  $Y$  are called extreme points of the convex hull. If  $Y$  is a finite set of points, the convex hull of  $Y$  can also be described using a finite set of hyperplanes in the objective space. Similar to Solanki et al. (1993), we use the following definitions.

**Definition 4.** A hyperplane in the objective space is given by  $H(w, b) = \{z \mid w^\top z = b\}$  with  $w \in \mathbb{R}^m \setminus \{0\}$  and  $b \in \mathbb{R}$ . The vector  $w$  is a normal of the hyperplane. If  $\|w\| = 1$ , the vector  $w$  is a unit normal.

**Definition 5.** The set  $HS(w, b) = \{z \mid w^\top z \geq b\}$  is the half-space given by  $w \in \mathbb{R}^m \setminus \{0\}$  and  $b \in \mathbb{R}$ . The vector  $w$  is an inner normal of the half-space. If  $\|w\| = 1$ , the vector  $w$  is the inner unit normal of the half-space. The vector  $\bar{w} = -w$  is an outer normal of the half-space.

In this paper, the vector  $w$  always refers to an inner unit normal unless explicitly specified otherwise.

**Definition 6.** If  $V \subset C$  where  $C$  is a convex set, then a hyperplane  $H(w, b)$  supports  $C$  at  $V$  if  $V \subset H(w, b)$  and  $C \subset HS(w, b)$ .

**Definition 7.** A set of points  $F$  is a  $k$ -face of  $C$  if  $F$  has dimensionality  $k$  and if there exists a supporting hyperplane  $H(w, b)$  which supports  $C$  at  $F$  and for which holds that  $H(w, b) \cap C = F$ . If  $C \subset \mathbb{R}^m$ , its  $(m - 1)$ -faces are facets and its 0-faces are the extreme points.

In this paper, we generally consider convex sets  $C$  with a finite number of facets. We denote the  $n$  facets belonging to a certain convex set  $C$  by  $F^1, \dots, F^n$  and the set of extreme points by  $C^E$ . When  $H(w^i, b^i)$ ,  $i = 1, \dots, n$ , are their supporting hyperplanes, then every point  $z \in C$  must satisfy  $(w^i)^\top z \geq b^i$  for  $i = 1, \dots, n$ . If  $z$  satisfies any of these inequalities with equality then  $z$  is on the corresponding facet. By taking the intersection of the half spaces  $HS(w^i, b^i)$ ,  $i = 1, \dots, n$ , defined by the facets, we can describe the convex hull as an intersection of half spaces.

To determine the convex hull, we use the function "convhulln" in Matlab which uses the Qhull algorithm (Barber et al. (1996)). For convex hulls of dimensions at least five, the output of this function may contain facets with empty area or volume. These facets are removed as they are redundant.

In all algorithms used in this paper, we approximate the Pareto set by a set of faces of  $\text{conv}\{Y\}$  with  $Y \subset Z$ . For this set  $Y$  it holds that  $\text{conv}\{Y\} = \text{conv}\{Y^E\}$  and that we know the corresponding  $x \in X$  for all points  $z \in Y^E$ . However, we do not know the corresponding  $x \in X$  for all other vectors  $z \in \text{conv}\{Y\}$ . For these vectors, we can use the following method to determine a vector  $x^* \in X$  such that  $z^* \leq z$  for  $z^* = f(x^*)$ . As  $z \in \text{conv}\{Y\}$ , the vector  $z$  can be written as a convex combination of  $m$  vectors in  $Y^E$ . Let us denote these vectors by  $z^1, \dots, z^m$  and the weights defining the convex combination by  $\lambda^1, \dots, \lambda^m$ . If  $x^1, \dots, x^m$  are the (known) vectors such that  $z^i = f(x^i)$  for  $i = 1, \dots, m$ , then we can define  $x^* = \sum_{i=1}^m \lambda^i x^i$ . Because of the convexity of  $X$  and  $f(x)$ , it is now easy to show that  $x^* \in X$  and  $z^* \leq z$  for  $z^* = f(x^*)$ .



### 3 Sandwich algorithms

#### 3.1 Inner and outer approximations

In the introduction, we mentioned that we want to find an accurate approximation of the complete relevant part of the Pareto set with as few optimizations as possible. Furthermore, we would also like to be able to give quality guarantees on the accuracy of the approximation. For both aims, sandwich algorithms seem to be very suitable. The main characteristic of sandwich algorithms is that they provide two approximations between which the set  $PS$  of Pareto optimal points is sandwiched. In the case of an MOP where all objectives must be minimized, these two approximations can more formally be defined as follows.

**Definition 8.** A set  $IPS \subseteq Z$  is an inner approximation of  $PS$  if it satisfies  $IPS \subseteq PS + \mathbb{R}_{\geq}^m$ .

**Definition 9.** A set  $OPS \subseteq Z$  is an outer approximation of  $PS$  if it satisfies  $PS \subseteq OPS + \mathbb{R}_{\geq}^m$ .

The inner and outer approximations are sometimes also called the upper and lower bounds or approximations. However, to avoid confusion with other upper and lower bounds, we choose to use the terms inner and outer approximation.

As the real Pareto set lies between the inner and outer approximation, we can use them to determine an upper bound on the approximation error. This upper bound can be used to steer the algorithm and to give guarantees. Steering can be done by generating new points in areas of the Pareto set where the upper bound on the approximation error is still relatively large. This way, we are likely to gain more in terms of accuracy than by generating points in areas where the upper bound is much smaller. By generating new points where the potential for improvement is highest, sandwich algorithms try to accurately approximate a Pareto set with as few optimization runs as possible. Furthermore, the upper bound on the approximation error gives the decision maker the guarantee that by using the approximation, he will never select a solution of which the objective vector  $z$  is more than a certain amount worse than an objective vector  $\tilde{z}$  which is an element of the real Pareto set.

In the next three sections, we describe four different sandwich methods.

#### 3.2 Algorithm of Solanki et al.

The main steps of the XNISE1 algorithm in Solanki et al. (1993) are:

1. Find all anchor points  $z^{A1}, \dots, z^{Am}$ , and all points  $z^{Mi} = \arg \max\{z_i \mid z \in PS\}$  for  $i = 1, \dots, m$ . Set  $IPS = \text{conv}\{z^{A1}, \dots, z^{Am}, z^{M1}, \dots, z^{Mm}\}$ .
2. Set  $OPS = \{z \mid z_i^{Ai} \leq z_i \leq z_i^{Mi}, i = 1 \dots, m\}$ .
3. Calculate the error for each facet (see below).
4. Select the facet  $F^*$  with the largest error. If this error is below a certain value, we stop. Otherwise, let  $H(w, b)$  be a supporting hyperplane at  $F^*$  and go to Step 5.
5. Determine  $z^*$  by solving:

$$z^* = \arg \min\{w^\top z \mid z \in Z\}.$$

6. If  $w^\top z^* = b$ , then no new extreme point is found. Set the error of this facet to zero and return to Step 4 if this happens. Otherwise, go to Step 7.
7. Update  $IPS$  by replacing it with  $\text{conv}\{z^*, IPS\}$ .
8. Update  $OPS$  by adding the inequality  $w^\top z \leq b$ .
9. Return to Step 3.

In Step 3, the error of the facet is calculated by first solving the following problem:

$$w^\top \bar{z} = \min\{w^\top z \mid z \in OPS\}. \quad (3)$$

Notice that this is just an LP-problem because  $OPS$  can be described by a set of linear inequalities. The error is calculated as the distance between  $\bar{z}$  and the hyperplane defined by the facet. As  $w$  satisfies  $\|w\| = 1$ , this distance can easily be calculated by  $b - w^\top \bar{z}$ .

Determining  $z^{M1}, \dots, z^{Mn}$  in Step 1 is not straightforward as we have no explicit description of  $PS$  and there exist no suitable weight vectors to determine these points with the weighted sum method. Moreover, the maximal values of the individual objectives can be attained at one or more of the anchor points. For example, the maximum of objective  $z_1$  can be attained by the anchor points  $z^{A2}$  or  $z^{A3}$ . For these reasons, we choose to change the algorithm by leaving out  $z^{M1}, \dots, z^{Mn}$  in Step 1 and replacing  $z_i^{Mi}$  by  $z_i^{ub}$  in Step 2.

The set  $IPS$  obtained with this algorithm forms an inner approximation of  $PS$ . However, not all points of this convex hull are useful as an approximation of  $PS$  as some are (weakly) dominated by other points in  $IPS$ . Solanki et al. (1993) developed the XNISE2 algorithm to remove all points in  $IPS$  which are weakly dominated by other points in  $IPS$ . The remaining set is used as the final approximation.

As mentioned by Solanki et al. (1993), taking the weight vector  $w$  equal to the normal of a facet can be problematic when the normal contains both positive and negative elements. In that case, minimizing the weighted sum in Step 5 could result in a non-Pareto optimal point, i.e., a point not part of  $PS$ . To avoid the found point to be too far away from  $PS$ , upper and lower bounds are put on the objective values in the approach of Solanki et al. (1993). In Sections 3.4 and 4, we see other methods of dealing with this situation.

### 3.3 Algorithm of Klamroth et al.

The algorithm of Klamroth et al. (2002) uses two separate algorithms for generating the inner and outer approximation of the Pareto set. They combine the two algorithms by alternatingly performing one iteration of each algorithm.

The algorithm for the inner approximation also starts with a facet defined by the anchor points. For this facet, the algorithm tries to find the point in the Pareto set furthest away from this facet using the so-called gauge-method. The gauge method determines a new point  $z^*$  by solving the following problem:

$$\begin{aligned} \gamma(z^*) &= \max \sum_{i=1}^k \lambda_i \\ \text{s.t.} \quad & z \leq z^{pN} + \sum_{i=1}^k \lambda_i (z^i - z^{pN}) \end{aligned} \quad (4)$$

$$\begin{aligned}\lambda_i &\geq 0 \text{ for } i = 1, \dots, k \\ z &\in Z,\end{aligned}$$

where the vectors  $z^1, \dots, z^k$  are the extreme points of the facet. The deviation  $dev(z^*)$  of the new point  $z^*$  is subsequently calculated by  $|\gamma(z^*) - 1|$ . The new point with the largest deviation is added to the inner approximation and new facets are determined by updating the convex hull. For all new facets the gauge method is performed and again the point with the largest deviation is added. This last step is repeated until some stopping criterion is satisfied.

This inner approximation method of Klamroth et al. (2002) is based on earlier research of Schandl et al. (2002), who formulate the gauge method slightly differently. In their formulation, the first inequality constraint in Problem 4 is formulated as an equality constraint. We tested both methods and refer to them as  $\text{Klamroth}^{\leq}$  and  $\text{Klamroth}^=$ , respectively.

Tests with both methods showed that in certain situations, the point with the largest deviation can be a point which is already part of the *IPS*. If this happens, the same point is added in every subsequent step of the algorithm and the *IPS* no longer changes. To avoid this behavior, we add a tabu-list to the algorithm. If applying the gauge-method for a certain facet results in an already found point, the facet is added to the tabu-list and is no longer evaluated in subsequent steps of the algorithm.

The outer approximation method is formulated as follows. First the utopia and pseudo-nadir point are determined. The hypercube defined by the utopia and pseudo-nadir point is taken as the initial outer approximation. The outer approximation is thus a convex polytope. We call the extreme points of this polytope the fundamental vectors and denote them by  $v^1, \dots, v^k$ . Next, we solve for every fundamental vector  $v^i \neq z^{pN}$  the problem:

$$\begin{aligned}\delta^i &= \max \lambda \\ \text{s.t.} \quad & f(x) - (z^{pN} + \lambda(v^i - z^{pN})) \leq 0 \\ & \lambda \geq 0 \\ & x \in X\end{aligned} \tag{5}$$

and denote the optimal Lagrange multiplier of the first constraint by  $u^i$  and the optimal value of  $f(x)$  by  $z^i$ . The fundamental vector with minimal value of  $\delta^i$  is now used to update the outer approximation. If  $k$  is the index of this vector, then the outer approximation is updated by adding the inequality  $(u^k)^\top z \geq (u^k)^\top z^k$ . After determining the fundamental vectors of this updated polytope, we solve problem (5) for all new vectors and again update the outer approximation using the vector with the largest  $\delta^i$ . This last step is repeated until some stopping criterion is satisfied.

### 3.4 Algorithm of Craft et al.

In Craft et al. (2006), the PGEN algorithm is introduced to approximate a convex Pareto set. The algorithm is similar to the algorithm of Solanki et al. (1993) but has two important differences.

The first difference is in the way the error is calculated. Instead of solving an LP-problem, the error is calculated by looking at the hyperplanes of the outer approximation going through the corner points of a facet. If these planes intersect in a point, this point is called the lower distal point. The error is defined as the distance between the facet and its lower distal point.

Secondly, the algorithm differs in the way it deals with facets having a normal with both positive and negative elements. Whereas Solanki et al. (1993) choose to put upper and lower bounds on the objective values, the PGEN algorithm deals with this problem by using a different weight vector. Instead of using the normal of the facet, a linear combination is taken of the weight vectors used to obtain the corner points of the facet.

## 4 Adding dummy points to $IPS$

### 4.1 General idea behind dummy points

As we mentioned in the previous section, the weighted sum method, used in both the algorithms of Solanki et al. (1993) and Craft et al. (2006), can give non-Pareto points if the used weight vector has both positive and negative components. However, facets of the convex hull may have normals with this property. Using these 'undesirable' normals as weight vectors can thus result in non-Pareto points. Solanki et al. (1993) and Craft et al. (2006) both developed different ways of dealing with this problem as described in Section 3.4. The drawback of the approach of Solanki et al. (1993) is that it can still produce non-Pareto points, although there is a limit on how far the non-Pareto point is from  $PS$ . The approach of Craft et al. (2006) does not have this drawback as it always uses a non-negative weight vector. However, tests with this algorithm show that sometimes the same facet with 'undesirable' normal is selected in subsequent iterations. This implies that solving the weighted sum problem for the alternative weight vector does not always produce a point which reduces the error measure used by Craft et al. (2006). As we assume that every optimization is time-consuming, this is not a desirable property.

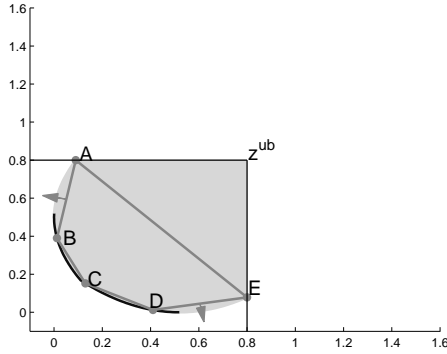


Figure 1: Example of facets with 'undesirable' normals.

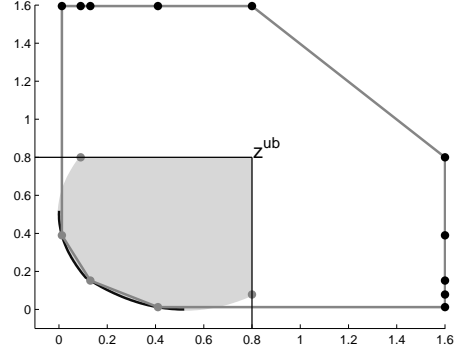


Figure 2: Example of avoiding facets with 'undesirable' normals by adding dummy points.

In this section, we introduce a new way of dealing with the problem of 'undesirable' normals. By taking the convex hull of  $IPS$  and a set of dummy points, we make sure that all relevant facets have a 'desirable' inner normal with only non-negative elements. Besides solving the problem of obtaining non-Pareto points, this approach has a number of other benefits. In Section 4.3, we will show that the dummy points help us in determining the points of  $IPS$  which are not dominated by other points of  $IPS$ . Furthermore, we introduce an error measure in Section 5 which has a number of desirable properties. One of these

properties is that calculating an upper bound for this measure based on  $IPS$  and  $OPS$  can be done by solving a number of LP-problems when using dummy points.

To explain the general idea behind these dummy points and the advantages, we use the bi-criteria example in Figure 1. Although the problem with 'undesirable' normals does not occur for bi-criteria problems, we use a bi-criteria example to make the explanation of the general idea behind the dummy points easier. In Figure 1, the shaded area represents  $Z$  and the points A, B, C, D, and E are the current extreme points of  $IPS$ . As the arrows indicate, facets AB and DE both have 'undesirable' normals. Using these normals in the weighted sum method means that we search in the direction of the arrows for the point furthest away from the facet. This would in both cases result in a non-Pareto solution. In Figure 2, two dummy points are added for each extreme point  $z \in IPS^E$ . The dummy points are created by replacing one of the two coordinates of  $z$  by a large value (An exact definition of the dummy points is given in Section 4.2). Once the dummy points are created, the set  $IPS$  is replaced by the convex hull of  $IPS$  and all dummy points. All facets containing at least one  $IPS$ -point now have a normal with only non-negative elements. All other facets, containing only dummy points, do not satisfy the upper bound constraint on the objectives and are thus not relevant.

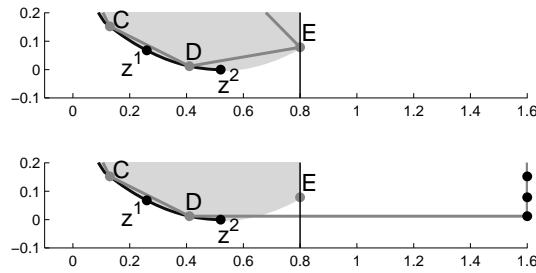


Figure 3: Example of improved error calculation with dummy points.

Besides solving the problem of 'undesirable' normals, adding dummy points also helps in determining the approximation error. To explain this, we use the points  $z^1$  and  $z^2$  in Figure 3. When the error in a certain point is calculated by the Euclidean distance to the approximation, point  $z^1$  will be considered more accurately approximated than point  $z^2$ . However, point D approximates  $z^2$  quite closely as it is only slightly worse in objective 2 and better in objective 1. The distance between  $z^2$  and the facet between D and the corresponding dummy point is therefore a better error measure. Now point  $z^2$  has a slightly better error value than  $z^1$  which seems a much better representation of the approximation accuracy. Adding the dummy points to the approximation thus also improves the usefulness of the Euclidean distance as an error measure. In Section 5, we introduce  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$  to measure the approximation error and show how we can easily calculate this measure using the dummy points.

Finally, the  $IPS$  obtained by any of the sandwich algorithms may contain points which are weakly or strongly dominated by other points in  $IPS$ . As it is not optimal for the decision maker to choose any of these points, we want to detect and remove these points before presenting  $IPS$  to the decision maker. As mentioned before, Solanki et al. (1993) presented the XNISE2 algorithm for this purpose. However when using dummy points, we can use an easier approach for finding these points. We introduce and discuss this new approach in Section 4.3.

## 4.2 Effect of dummy points on inner normals

To prove that the above ideas also work for higher dimensions, we first give a formal definition of the dummy points. For a fixed and small  $\theta > 0$  and for each extreme point  $z \in IPS^E$ , the dummy points  $d^1(z), \dots, d^m(z)$  are defined in the following way.

**Definition 10.** Let  $\theta > 0$ . The dummy point  $d^i(z)$  generated from  $z \in Z$  is given by:

$$d_j^i(z) = \begin{cases} z_j & \text{if } j \neq i \\ mz_j^{ub} + \theta & \text{if } j = i \end{cases} \quad \text{for } j = 1, \dots, m.$$

**Definition 11.** The set  $D$  is defined as:

$$D = \{d^i(z) \mid z \in IPS^E, i = 1, \dots, m\}.$$

Note that all dummy points are weakly dominated by the points they are generated from. They will thus never dominate any of the Pareto optimal points in  $IPS$ . Furthermore, dummy points can, in many instances, be written as a convex combination of other dummy points. In the example of Section 4.1, this is true for six of the ten generated dummy points. The points for which this happens can easily be detected. As we always use  $D$  to determine  $\text{conv}\{IPS, D\}$ , these points can be removed from  $D$ , which reduces the computation costs of determining  $\text{conv}\{IPS, D\}$ . Lastly, because the dummy points are not smaller or equal to  $z^{ub}$ , certain facets of  $\text{conv}\{IPS, D\}$  may contain no points  $\tilde{z}$  for which  $\tilde{z} \leq z^{ub}$ . As all points  $z \in PS$  do satisfy  $z \leq z^{ub}$ , these facets are not relevant as an approximation of  $PS$ . In the example in the previous section, we thus have only four relevant facets. To distinguish relevant from irrelevant facets, we introduce the following formal definitions of relevant facets and hyperplanes.

**Definition 12.** A facet  $F$  of  $\text{conv}\{IPS, D\}$  is a relevant facet if at least one of its extreme points is in  $IPS$ . The set of all relevant facets is denoted by  $RF$ .

**Definition 13.** A hyperplane  $H(w, b)$  is a relevant hyperplane if it supports  $\text{conv}\{IPS, D\}$  at a relevant facet. The set of all relevant hyperplanes is denoted by  $RH$ .

As we assume that  $z \leq z^{ub}$  for every  $z \in IPS$ , all relevant facets contain points which are smaller or equal than  $z^{ub}$  and can thus approximate  $PS$ . Lemma 1 shows that all facets of  $\text{conv}\{IPS, D\}$  not satisfying the above definition are not relevant.

**Lemma 1.** Consider a facet  $F$  of  $\text{conv}\{IPS, D\}$ . If  $F$  is not a relevant facet, then there is no point  $z$  on  $F$  with  $z \leq z^{ub}$ .

*Proof.* As  $F$  is not a relevant facet, all extreme point must be dummy points. Furthermore, any point  $z$  on  $F$  can be written as  $z = \sum_{i=1}^m \lambda_i z^i$  with  $z^1, \dots, z^m$  extreme points of  $F$ ,  $\sum_{i=1}^m \lambda_i = 1$  and  $\lambda_i \geq 0$  for  $i = 1, \dots, m$ . Assume without loss of generality that  $\lambda_1 = \max_i \lambda_i$ . From  $\sum_{i=1}^m \lambda_i = 1$  and  $\lambda_i \geq 0$  for  $i = 1, \dots, m$  follows that  $\lambda_1 \geq \frac{1}{m}$ . As  $z^i \geq 0$  for all  $i = 1, \dots, m$ , this gives the following inequality:

$$z = \lambda_1 z^1 + \sum_{i=2}^m \lambda_i z^i \geq \lambda_1 z^1 \geq \frac{1}{m} z^1.$$

As  $z^1$  is a dummy point,  $z_j^1 = mz_j^{ub} + \theta$  must hold for one element  $j$ . Combining this with the found inequality gives:

$$z_j \geq \frac{1}{m}z_j^1 = \frac{1}{m}(mz_j^{ub} + \theta) > z_j^{ub}.$$

This shows that there is no point on  $F$  for which  $z \leq z^{ub}$ .  $\square$

In Section 4.1, we already mentioned that by using  $\text{conv}\{IPS, D\}$  instead of  $IPS$ , we want to make sure that all relevant facets have an inner normal with only non-negative elements. Lemma 2 shows that for the above defined relevant facets and set  $D$  this indeed holds.

**Lemma 2.** *Consider  $\text{conv}\{IPS, D\}$  and  $F \in RF$ . If  $H(w, b) \in RH$  is a supporting hyperplane at  $F$ , then  $w \geq 0$ .*

*Proof.* Let  $z$  be an extreme point of  $F$  that is in  $IPS$ . Suppose by contradiction that there is an  $i$  such that  $w_i < 0$ . Then for the dummy point  $d^i(z)$  it holds:

$$w^\top d^i(z) = w^\top z + w_i(mz_i^{ub} + \theta - z_i) < w^\top z = b.$$

This inequality is a contradiction because  $d^i(z)$  is in  $\text{conv}\{IPS, D\}$  which has  $H(w, b)$  as a supporting hyperplane.  $\square$

### 4.3 Determining non-IPS-dominated points of IPS

When we have determined an inner approximation  $IPS$  of  $PS$ , some points in  $IPS$  may be dominated by other points in  $IPS$ . For a decision maker it is not optimal to choose one of these points from  $IPS$ . Therefore, we want to determine all points in  $IPS$  which are not strongly or weakly dominated by other points in  $IPS$ . We will refer to these points as non-IPS-dominated points and denote the set of all non-IPS-dominated points by  $IPS^{nId}$ . The set  $\text{conv}\{IPS, D\}$  and the following lemma and proposition can be used to determine  $IPS^{nId}$ .

**Lemma 3.** *Consider  $z \in \text{conv}\{IPS, D\}$  with  $z \notin IPS$ . Then  $z$  is weakly dominated by a point in  $IPS$ .*

*Proof.* Let us denote all points in  $IPS^E$  by  $z^1, \dots, z^k$ . As  $z \in \text{conv}\{IPS, D\}$ , we know that  $z$  can be written as  $z = \sum_i \lambda_i z^i + \sum_{i,j} \mu_{i,j} d^j(z^i)$  with  $\sum_i \lambda_i + \sum_{i,j} \mu_{i,j} = 1$ ,  $\lambda_i \geq 0$  and  $\mu_{i,j} \geq 0$  for  $i = 1, \dots, k$ ,  $j = 1, \dots, m$ . Furthermore, because  $z \notin IPS$ ,  $\mu_{i,j} > 0$  must hold for at least one combination of  $i$  and  $j$ . By definition, each dummy point  $d^j(z^i) \in D$  is weakly dominated by the point  $z^i \in IPS^E$  it is generated from. Define  $z^* = \sum_i \lambda_i z^i + \sum_{i,j} \mu_{i,j} z^i$  as the point obtained by replacing each dummy point  $d^j(z^i)$  with  $z^i$  in the convex combination which formed  $z$ . This point  $z^*$  is an element of  $IPS$  and weakly dominates  $z$ .  $\square$

The above lemma thus shows we only have to consider points  $z \in IPS$ , even when using  $z \in \text{conv}\{IPS, D\}$  as an inner approximation.

**Proposition 1.** *Consider  $z^{IPS} \in IPS$ . Then  $z^{IPS}$  is not strongly dominated by another point in  $IPS$  if and only if  $z^{IPS}$  is on a relevant facet of  $\text{conv}\{IPS, D\}$ .*

*Proof.* Assume that  $z^{IPS}$  is not strongly dominated by another point in  $IPS$ . This implies that there exists no  $\hat{z} \in \text{conv}\{IPS, D\}$  such that  $\hat{z} < z^{IPS}$  and that  $z^{IPS}$  should thus be part of a facet  $F$  of  $\text{conv}\{IPS, D\}$ . As  $z^{IPS} \in IPS$ ,  $z^{IPS} \leq z^{ub}$  must hold and, according to Lemma 1, facet  $F$  should be a relevant facet of  $\text{conv}\{IPS, D\}$ .

Let us now assume that  $z^{IPS}$  is on a relevant facet  $F$  of  $\text{conv}\{IPS, D\}$  and, by contradiction, that there would exist a point  $\tilde{z} \in IPS$  such that  $\tilde{z}$  strongly dominates  $z^{IPS}$ . If  $H(w, b)$  is a supporting hyperplane at  $F$ , all points  $z \in \text{conv}\{IPS, D\}$  must satisfy  $w^\top z \geq b$ . Furthermore, we know  $w^\top z^{IPS} = b$  and  $w \geq 0$  by Lemma 2. From this it follows:

$$w^\top \tilde{z} < w^\top z^{IPS} = b.$$

This inequality shows that  $\tilde{z}$  cannot be an element of  $IPS$ . By contradiction, we have thus shown that  $z^{IPS}$  cannot be strongly dominated by another point in  $IPS$ .  $\square$

By taking the points on the relevant facets of  $\text{conv}\{IPS, D\}$  which are also in  $IPS$ , we can obtain a set of points in  $IPS$  which are not strongly dominated by any other point in  $IPS$ . The added dummy points thus make it quite easy to detect and discard the strongly dominated points. However, the set can still contain points which are weakly dominated by a point in  $IPS$ . Take for instance the points  $z^1 = (1, 0, 1)^\top$ ,  $z^2 = (0, 1, 1)^\top$  and  $z^3 = (0.5, 0.5, 0)^\top$ . For an MOP with three objectives, these points could be elements of  $IPS$  and be on a facet of  $\text{conv}\{IPS, D\}$ . However, then the point  $\tilde{z} = 0.5z^1 + 0.5z^2 = (0.5, 0.5, 1)^\top$  also has these properties, but is weakly dominated by  $z^3$ . Therefore, we need a different criterion to discard the weakly dominated points. The following proposition provides such a criterion.

**Proposition 2.** Consider  $z^{IPS} \in IPS$ . Let  $W$  be the set of inner unit normals of the relevant facets of  $\text{conv}\{IPS, D\}$  containing  $z^{IPS}$  and let  $w^+$  be the sum of all  $w \in W$ . Then  $z^{IPS}$  is not weakly dominated by a point in  $\text{conv}\{IPS, D\}$  if and only if  $w^+ > 0$ .

*Proof.* Assume  $w^+ > 0$ . This implies that for each dimension  $d = 1, \dots, m$ , there exists an inner normal  $w^d \in W$  for which it holds  $w_d^d > 0$ . Let  $H(w^d, b^d)$  be the corresponding supporting hyperplane of the facet, so that, by definition,  $w^{d\top} z \geq b^d$  for  $z \in \text{conv}\{IPS, D\}$  and  $w^{d\top} z^{IPS} = b^d$ . As  $w^d$  is an inner normal of a relevant facet,  $w^d \geq 0$  according to Lemma 2. If  $\hat{z}$  is a point which (weakly) dominates  $z^{IPS}$  then there is a dimension  $d$  such that  $\hat{z}_d < z_d^{IPS}$ . Because  $w^d \geq 0$  and  $w_d^d > 0$ , it follows that  $w^{d\top} \hat{z} < w^{d\top} z^{IPS} = b^d$ , so that  $\hat{z}$  is not an element of  $\text{conv}\{IPS, D\}$ . Since  $w^+ > 0$ , there thus exists no  $\hat{z}$  in  $\text{conv}\{IPS, D\}$  which weakly dominates  $z^{IPS}$ .

Let us now assume that  $z^{IPS}$  is not weakly dominated by a point in  $\text{conv}\{IPS, D\}$ . This implies that any point  $\tilde{z}$  satisfying  $\tilde{z}_d < z_d^{IPS}$  and  $\tilde{z}_i = z_i^{IPS}$  for  $i \neq d$  is not in  $\text{conv}\{IPS, D\}$ . As this holds for any such point  $\tilde{z}$ , there must exist a supporting hyperplane  $H(w^d, b^d)$  of  $\text{conv}\{IPS, D\}$  such that  $w^{d\top} z^{IPS} = b^d$  and  $w^{d\top} \tilde{z} < b^d$ . Because  $\tilde{z}_d - z_d^{IPS} < 0$  and

$$w_d^d(\tilde{z}_d - z_d^{IPS}) = w^{d\top}(\tilde{z} - z^{IPS}) < b^d - b^d = 0,$$

it follows that  $w_d^d > 0$ . The vector  $w^d$  must be in  $W$  as only relevant facets can contain points  $z^{IPS} \leq z^{ub}$ . Furthermore, as  $w \geq 0$  for all  $w \in W$  and  $w_d^d > 0$ ,  $w_d^+$  must also be greater than zero. As this holds for every dimension  $d = 1, \dots, m$ , we have got  $w^+ > 0$ .  $\square$

To determine  $IPS^{nId}$ , we can check all points on relevant facets of  $IPS$  with the criterion in Proposition 2. However as the relevant facets of  $IPS$  contain infinitely many points, this



requires a more efficient method. The set  $W$  used in Proposition 2, however, is the same for all points on a face of  $IPS$ . This implies that we can check all points on a face simultaneously. If  $IPS$  is the convex hull of a finite number of points, the number of faces of  $IPS$  is also finite. As this is true for all previously discussed sandwich algorithms, we can use the following algorithm, inspired by XNISE2 of Solanki et al. (1993), to determine  $IPS^{nId}$ :

1. Set  $d = m$  and  $IPS^{nId} = \emptyset$ .
2. Denote by  $P^d$  the set of all  $(d - 1)$ -faces of  $\text{conv}\{IPS, D\}$  having  $d$  extreme points in  $IPS$ .
3. For each face in  $P^d$ , determine if it is a subset of a higher dimensional face in  $IPS^{nId}$ . If so, remove the face from  $P^d$ .
4. For each remaining face in  $P^d$ , calculate the vector  $w^+$  by taking the sum of the inner unit normals of the facets of which this  $(d - 1)$ -face is a subset. If  $w^+ > 0$ , add the face to  $IPS^{nId}$ .
5. Set  $d = d - 1$ .
6. If  $d \geq 1$ , return to Step 2. Otherwise, stop.

To illustrate the above algorithm, we use the example given in Figure 4. When we take  $IPS = \text{conv}\{z^1, z^2, z^3, z^4\}$  and  $z^{ub} = [1 \ 1 \ 1]^\top$ , the figure shows all relevant facets of  $\text{conv}\{IPS, D\}$ . The Pareto optimal points are on the facet with extreme points  $z^1$ ,  $z^2$ , and  $z^3$ , and on the edge with extreme points  $z^1$  and  $z^4$ . In the algorithm, we look at the 2-, 1- and 0-faces which in three dimensions correspond to facets, edges and extreme points, respectively.

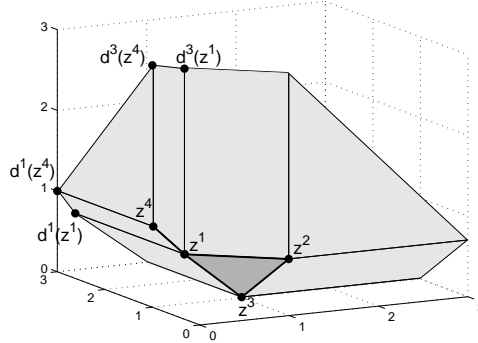


Figure 4: Example of determining non- $IPS$ -dominated points in  $IPS$ .

We start by looking at all facets in the set  $P^3$ , which in this example only contains the facet with extreme points  $z^1$ ,  $z^2$ , and  $z^3$ . As the inner normal  $w$  of this facet satisfies  $w > 0$ , we have  $w^+ > 0$  and the facet is added to  $IPS^{nId}$  in Step 4. Next, we take  $d = 2$  and continue with the set  $P^2$  containing all edges of  $\text{conv}\{IPS, D\}$  connecting two extreme points of  $IPS$ . This implies that  $P^2 = \{(z^1, z^2), (z^1, z^3), (z^2, z^3), (z^1, z^4)\}$ . As the first three edges are part of the facet which we already added to  $IPS^{nId}$ , they are removed in Step 3. For the remaining edge  $(z^1, z^4)$ , we determine vector  $w^+$  by summing the inner normals of the facets defined by  $\{z^1, z^4, d^1(z^4), d^1(z^1)\}$  and  $\{z^1, z^4, d^3(z^4), d^3(z^1)\}$ . The first facet has an inner unit normal  $w^1$

with  $w_1^1 = 0$  and  $w_2^1, w_3^1 > 0$ . For the inner unit normal  $w^2$  of the second facet it holds  $w_3^2 = 0$  and  $w_1^2, w_2^2 > 0$ . For all points on the edge  $(z^1, z^4)$ , the vector  $w^+$  is equal to  $w^1 + w^2 > 0$ . The edge  $(z^1, z^4)$  should thus be added to the set  $IPS^{nId}$  according to Proposition 2 and this is exactly what the algorithm does. Finally, we look at all points in  $P^1 = \{z^1, z^2, z^3, z^4\}$ . As all these point are already in  $IPS^{nId}$ , the set  $P^1$  becomes empty in Step 3 and the algorithm is finished.

## 5 Error measure

### 5.1 Motivation and definition of $\alpha(PS, IPS)$ and $\alpha(OPS, IPS)$

To assess the quality of different outer and inner approximations, we need a measure to quantify the accuracy of the approximation. There exist many different measures and methods which can be used to determine the quality of an approximation. Carlyle et al. (2003) and Zitzler et al. (2003) provide two extensive reviews and analyses of different comparison methods. In most sandwich algorithms, the accuracy is measured by the distance between the inner approximation and the real Pareto set. However, there are differences in the exact definition of this distance and in the way it is calculated.

To decide which definition to use, we first have to look at how we define the accuracy of an approximation. Recall that we assume that the user is interested in the complete Pareto set. Therefore, we could call an approximation accurate if each point of the Pareto set is represented accurately by a point on the approximation. Note that this definition is essentially different from measuring if each point on the approximation accurately represents a point of the Pareto set. Consider, for instance, an approximation which consists of only one point which is an element of the Pareto set  $PS$ . According to the second definition, this would be a very accurate approximation as all points of the approximation are in  $PS$ . However, using the first definition, the approximation is probably not very accurate as this one point is not likely to be an accurate approximation of all points in  $PS$ . Consider, on the other hand, an approximation which consists of all points in  $PS$  plus a point which does not accurately represent a point in  $PS$ . Now the approximation is very accurate according to the first definition but not according to the second. As these examples illustrate, we could say that the first definition determines if an approximation is not too small and the second if it is not too large. Ideally, a measure should be such that it indicates when an approximation is accurate according to both definitions.

When using these definitions, we still have to specify when a point in  $PS$  is accurately represented and when a point of the approximation is an accurate representation of a point in  $PS$ . To decide on this, we also take into account the final user of the approximation. In our opinion, the measure should not only measure the accuracy in an adequate way but should also be easy to understand by the user. The second aspect implies that the interpretation of the measure should be easy to explain. This is not only important because it helps the user in accepting the approximation, but also because the user must be able to specify a desired accuracy. We assume that most users prefer to specify their desired accuracy in terms of a maximum percentage or absolute amount of allowed inaccuracy per objective.

Taking this into account,  $\epsilon$ -dominance seems to be a suitable way to measure whether a point in the Pareto set is accurately represented. This measure exists in two different variations: additive and multiplicative. The additive variant was simultaneously introduced by Evtushenko and Potapov (1987) and Reuter (1990) and the multiplicative variant by Ruhe

and Fruhwirth (1990). The additive  $\epsilon$ -dominance is defined as follows.

**Definition 14.** Let  $\epsilon \in \mathbb{R}_{\geq}^m$ . A point  $z$  is  $\epsilon$ -dominated by a point  $\hat{z}$  if:

$$\hat{z} \leq z + \epsilon.$$

This definition implies that a point is  $\epsilon$ -dominated if there exists a point which is at most  $\epsilon_i$  worse in each objective  $i = 1, \dots, m$ . For multiplicative  $\epsilon$ -dominance,  $z + \epsilon$  is replaced by  $z(1 + \epsilon)$ , which implies that there must exist a point which is at most a factor  $\epsilon_i$  worse in each objective  $i = 1, \dots, m$ . An upper bound is thus set on either the absolute error or the relative error per objective. The choice which variant to use, depends mainly on the kind of guarantee the user prefers. Note however that using the multiplicative variant can be difficult if objective values can get close or equal to zero. In this paper, we therefore choose to use the additive variant.

When all points in  $PS$  are  $\epsilon$ -dominated by points in  $IPS$ , we call  $IPS$  an  $\epsilon$ -approximation. If  $IPS$  is not an  $\epsilon$ -approximation, we are interested to know how far off it is from an  $\epsilon$ -approximation. We can measure this by determining the smallest multiple of  $\epsilon$  for which  $PS$  is  $\epsilon$ -dominated. Therefore, we introduce the following definitions of  $\alpha(z, T)$  and  $\alpha(S, T)$ .

**Definition 15.** For a fixed  $\epsilon \in \mathbb{R}_{\geq}^m$ , the error measure  $\alpha(z, T)$  is the minimal  $\alpha$  for which point  $z$  is  $\alpha\epsilon$ -dominated by a point in the set  $T$ . For the set  $S$ , the measure  $\alpha(S, T)$  is defined as  $\max_{z \in S} \alpha(z, T)$ .

The value of  $\alpha(PS, IPS)$  is thus a measure of how close the approximation  $IPS$  is to being an  $\epsilon$ -approximation of  $PS$ . In Section 8, we will use the value of  $\alpha$  to compare different  $IPS$ s.

When we do not know  $PS$ , we can still use the inner and outer approximation of  $PS$  to determine upper bounds on the above accuracy measure. If all points in  $OPS$  are  $\epsilon$ -dominated by points in  $IPS$ , then also all points in  $PS$  must be  $\epsilon$ -dominated by points in  $IPS$ . Therefore,  $\alpha(OPS, IPS)$  is an upper bound for  $\alpha(PS, IPS)$ . Furthermore, note that  $\alpha(z, IPS) = \alpha(z, IPS^{nId})$ , which implies that  $IPS^{nId}$  is an  $\epsilon$ -approximation of  $PS$  if and only if  $IPS$  is an  $\epsilon$ -approximation of  $PS$ .

To determine whether a point of the approximation is an accurate approximation of a point in  $PS$ , we use a similar concept to  $\epsilon$ -dominance called  $\varepsilon$ -Pareto optimality or  $\varepsilon$ -efficiency. This concept was introduced by Loridan (1984) and is defined as follows.

**Definition 16.** Let  $\varepsilon \in \mathbb{R}_{\geq}^m$ . A point  $z \in Z$  is  $\varepsilon$ -Pareto optimal if there is no  $\hat{z} \in Z$  such that:

$$\hat{z} \leq z - \varepsilon.$$

This definition implies that a point is  $\varepsilon$ -Pareto optimal if there exists no point which is more than  $\varepsilon_i$  better in each objective  $i = 1, \dots, m$ .

To satisfy both definitions of accuracy discussed at the beginning of this section, we aim to find approximations  $IPS$  and  $OPS$  such that each point of  $PS$  is  $\epsilon$ -dominated by a point of  $IPS^{nId}$  and such that each point of  $IPS^{nId}$  is  $\varepsilon$ -Pareto optimal. As illustrated, an approximation which satisfies the first criterion not necessarily satisfies the second. However, if  $\epsilon = \varepsilon$ , we can show that all points of  $IPS^{nId}$  are  $\varepsilon$ -Pareto optimal when  $IPS^{nId}$  is an  $\epsilon$ -approximation. We prove this result in the following proposition.

**Proposition 3.** Let  $\epsilon \in \mathbb{R}_{\geq}^m$  and let  $IPS^{nId}$  be an  $\epsilon$ -approximation of  $PS$ . Then all points  $z \in IPS^{nId}$  are  $\epsilon$ -Pareto optimal.

*Proof.* Assume, by contradiction, that there exists a point  $\hat{z} \in PS$  such that  $\hat{z} \leq z - \epsilon$ . As  $IPS^{nId}$  is an  $\epsilon$ -approximation of  $PS$ , there must exist a  $\tilde{z} \in IPS^{nId}$  such that  $\tilde{z} \leq \hat{z} + \epsilon$ . Combining these inequalities gives  $\tilde{z} \leq z$ . As  $\tilde{z}$  and  $z$  are both elements of  $IPS$ , this implies that  $z$  cannot be an element of  $IPS^{nId}$  as it is (weakly) dominated by  $\tilde{z} \in IPS$ . This contradiction shows that there exists no point  $\hat{z} \in PS$  such that  $\hat{z} \leq z - \epsilon$  and hence that  $z$  is  $\epsilon$ -Pareto optimal.  $\square$

## 5.2 Calculating $\alpha(PS, IPS)$ and $\alpha(OPS, IPS)$ using $\text{conv}\{IPS, D\}$

In this section, we introduce a method to calculate  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$  for a fixed  $\epsilon \in \mathbb{R}_{>}^m$  when  $|RH|$  is finite. For all  $IPS$ s obtained by the discussed sandwich algorithms, this last condition is satisfied as  $IPS$  is the convex hull of a finite number of points and thus has a finite number of facets. One main advantage of the method introduced in this section is that  $\alpha(OPS, IPS)$  can be calculated by solving a number of LP-problems. These LP-problems are the same as the ones used in the algorithm of Solanki et al. (1993) to determine the error of a facet. As the proofs and reasoning for  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$  are the same, we give them only for the first.

The general idea behind the method is the following. We first prove that for each  $z \in PS$  and  $H(w, b) \in RH$  it holds that  $\alpha(z, IPS) \geq \alpha(z, H(w, b))$ . Furthermore, we show that for each  $z \in PS$  equality holds for at least one hyperplane in  $RH$ , which implies that:

$$\alpha(z, IPS) = \max_{H(w, b) \in RH} \alpha(z, H(w, b)). \quad (6)$$

Instead of solving the above problem for every  $z \in PS$ , we determine  $\max_{z \in PS} \alpha(z, H(w, b))$  for every  $H(w, b) \in RH$ . Thus, we have to solve  $|RH|$  weighted sum problems. Using that

$$\begin{aligned} \alpha(PS, IPS) = \max_{z \in PS} \alpha(z, IPS) &= \max_{z \in PS} \max_{H(w, b) \in RH} \alpha(z, H(w, b)) \\ &= \max_{H(w, b) \in RH} \max_{z \in PS} \alpha(z, H(w, b)), \end{aligned} \quad (7)$$

we can then determine  $\alpha(PS, IPS)$  by taking the maximum over all solutions of these weighted sum problems. As mentioned before, the weighted sum problems become LP-problems when calculating  $\alpha(OPS, IPS)$ .

To prove the different steps in the above algorithm, we first need to prove the following result.

**Lemma 4.** Let  $\epsilon \in \mathbb{R}_{>}^m$  and  $H(w, b) \in RH$ . For every  $z \in PS$ , we have that  $\alpha(z, H(w, b))$  is equal to the unique  $\hat{\alpha}$  for which the point  $z + \hat{\alpha}\epsilon$  is on the hyperplane  $H(w, b)$ .

*Proof.* The existence and uniqueness of  $\hat{\alpha}$  such that  $z + \hat{\alpha}\epsilon$  is on  $H(w, b)$  follows easily from the fact that  $w^\top \epsilon \neq 0$  because  $w \geq 0$  and  $\epsilon > 0$ . As  $z$  is  $\alpha\epsilon$ -dominated by  $z + \hat{\alpha}\epsilon \in H(w, b)$ , we know  $\alpha(z, H(w, b)) \leq \hat{\alpha}$ . If  $\alpha(z, H(w, b)) < \hat{\alpha}$ , there must exist a  $\hat{z} \in H(w, b)$  such that  $\hat{z} < z + \hat{\alpha}\epsilon$ . However, this would imply:

$$b = w^\top \hat{z} < w^\top (z + \hat{\alpha}\epsilon) = b.$$

Due to this contradiction, we can conclude that  $\alpha(z, H(w, b)) = \hat{\alpha}$ .  $\square$

Using the results of Lemmas 1, 2 and 4, we can prove the following two lemmas.

**Lemma 5.** Consider  $\epsilon \in \mathbb{R}_{>}^m$ ,  $z \in PS$  and  $H(w, b) \in RH$ . Then  $\alpha(z, IPS) \geq \alpha(z, H(w, b))$ .

*Proof.* According to Lemma 2,  $w \geq 0$ . Assume by contradiction that  $z$  is  $\hat{\alpha}\epsilon$ -dominated by  $IPS$  with  $\hat{\alpha} < \alpha(z, H(w, b))$ . This implies that there should exist a point  $\hat{z} \in IPS$  such that  $\hat{z} \leq z + \hat{\alpha}\epsilon$ . For this point  $\hat{z}$  it holds:

$$w^\top \hat{z} \leq w^\top z + \hat{\alpha}(w^\top \epsilon) < w^\top z + \alpha(z, H(w, b))(w^\top \epsilon) = w^\top (z + \alpha(z, H(w, b))\epsilon) = b,$$

where the last equality follows from Lemma 4. As  $H(w, b)$  is a supporting hyperplane of a relevant facet of  $\text{conv}\{IPS, D\}$ , for all points in  $IPS$  it must hold that  $w^\top z \geq b$ . This implies that  $\hat{z}$  cannot be an element of  $IPS$ . Therefore,  $z$  can only be  $\alpha\epsilon$ -dominated by  $IPS$  with  $\alpha \geq \alpha(z, H(w, b))$ .  $\square$

**Lemma 6.** Consider  $\epsilon \in \mathbb{R}_{>}^m$ ,  $z \in PS$ . Then  $\alpha(z, IPS) = \alpha(z, H(w, b))$  must hold for at least one  $H(w, b) \in RH$ .

*Proof.* Define  $z^\alpha = z + \alpha(z, IPS)\epsilon$  and let  $z^{IPS} \in IPS$  be such that  $\alpha(z, \{z^{IPS}\}) = \alpha(z, IPS)$ . By the definition of  $\alpha(z, \cdot)$ , we know that  $z^{IPS} \leq z^\alpha$  with equality for at least one coordinate. Furthermore, there cannot exist a point  $\hat{z} \in IPS$  such that  $\hat{z} < z^{IPS}$  or  $\hat{z} < z^\alpha$ . Because all points in  $D$  are (weakly) dominated by points in  $IPS$ , the same is true for all points  $\hat{z}$  in  $\text{conv}\{IPS, D\}$ . This implies that  $z^{IPS}$  must lie on one or more facets of  $\text{conv}\{IPS, D\}$ . Let  $F$  be one of these facets. Because  $z^{IPS} \leq z^{ub}$  and  $z^{IPS} \in F$ ,  $F$  must be a relevant facet of  $\text{conv}\{IPS, D\}$  according to Lemma 1. Let  $H(\tilde{w}, \tilde{b}) \in RH$  be a supporting hyperplane of  $F$ . For this hyperplane it holds  $z^{IPS} \in F \subset H(\tilde{w}, \tilde{b})$ . Using this information, we can show the following relation:

$$\alpha(z, H(\tilde{w}, \tilde{b})) \leq \alpha(z, \{z^{IPS}\}) = \alpha(z, IPS).$$

The inequality follows from the definition of  $\alpha(z, \cdot)$  and the fact that  $z^{IPS} \in H(\tilde{w}, \tilde{b})$ . The equality follows from the definition of  $z^{IPS}$ . Combining the above equation with Lemma 5 gives that  $\alpha(z, IPS) = \alpha(z, H(\tilde{w}, \tilde{b}))$  must hold for  $H(\tilde{w}, \tilde{b}) \in RH$ .  $\square$

By combining Lemmas 5 and 6, we obtain equation (6) which gives an expression for  $\alpha(z, IPS)$ . By definition, to determine  $\alpha(PS, IPS)$ , we must take the maximum of  $\alpha(z, IPS)$  over all  $z \in PS$ . However, determining  $\alpha(z, IPS)$  explicitly for every  $z \in PS$  is not an option as there are an infinite number of vectors in the set  $PS$ . The number of  $H(w, b) \in RH$ , on the other hand, is finite. Therefore, we decide to calculate  $\max_{z \in PS} \alpha(z, H(w, b))$  for every  $H(w, b) \in RH$  and use these results to calculate  $\alpha(PS, IPS)$ . Another difficulty is that an explicit description of the set  $PS$  is often not known. However, the set  $Z$  can be described using the objectives and constraints of the MOP. The following proposition shows that determining  $\max_{z \in PS} \alpha(z, H(w, b))$  can be done by solving a weighted sum problem over the set  $Z$ .

**Proposition 4.** Consider a hyperplane  $H(w, b) \in RH$ , a vector  $\epsilon \in \mathbb{R}_{>}^m$  and the two optimization problems:

$$\begin{aligned} \beta^* = \min \quad & w^\top z \\ \text{s.t.} \quad & z \in Z, \end{aligned} \tag{8}$$

and

$$\begin{aligned} \alpha^* = \max \quad & \alpha \\ \text{s.t.} \quad & z \in PS \\ & w^\top(z + \alpha\epsilon) = b. \end{aligned} \tag{9}$$

The values  $\alpha^*$  and  $\beta^*$  are related as  $\alpha^* = \frac{b-\beta^*}{w^\top\epsilon}$  and  $\alpha^*$  is equal to  $\max_{z \in PS} \alpha(z, H(w, b))$ .

*Proof.* As  $H(w, b) \in RH$ ,  $w \geq 0$  according to Lemma 2. Using this fact, we can easily show that Problem 8 always has at least one solution  $\hat{z}$  for which it holds that  $\hat{z} \in PS$ . This implies that replacing the constraint  $z \in Z$  by  $z \in PS$  gives the same value of  $\beta^*$ . Furthermore, we know from the proof of Lemma 4 that for every  $z \in PS$ , there exists a unique  $\alpha$  which satisfies  $w^\top(z + \alpha\epsilon) = b$ . Using these two observations, we can change the first problem into the following problem which gives the same value of  $\beta^*$ :

$$\begin{aligned} \beta^* = \min \quad & w^\top z \\ \text{s.t.} \quad & z \in PS \\ & w^\top(z + \alpha\epsilon) = b. \end{aligned}$$

Because  $w^\top\epsilon > 0$ , the last constraint can also be written as  $\alpha = \frac{b-w^\top z}{w^\top\epsilon}$ . This relation implies that a vector  $\hat{z} \in PS$  which maximizes  $w^\top z$  also maximizes  $\alpha$ . We thus obtain the relation  $\alpha^* = \frac{b-\beta^*}{w^\top\epsilon}$  between  $\alpha^*$  and  $\beta^*$ .

Lastly, for every combination of  $z$  and  $\alpha$  which satisfies the two constraints of problem (9) it holds that  $\alpha$  is equal to  $\alpha(z, H(w, b))$  according to Lemma 4. Therefore,  $\alpha^*$  is equal to  $\max_{z \in PS} \alpha(z, H(w, b))$ .  $\square$

By solving the above problem for every  $H(w, b) \in RH$ , we can thus determine the value of  $\max_{z \in PS} \alpha(z, H(w, b))$  explicitly for every  $H(w, b) \in RH$ . Equation (7) shows that taking the maximum over all these solutions gives  $\alpha(PS, IPS)$ .

By replacing  $Z$  and  $PS$  by  $OPS$  in the above equations and lemmas, we obtain a similar method for determining  $\alpha(OPS, IPS)$ . Notice, that problems (3) and (8) then become the same. Calculating  $\alpha(OPS, IPS)$  thus requires no more effort than calculating the error measure of Solanki et al. (1993). However, the advantage of using  $\alpha(OPS, IPS)$  is that we can give an interpretation of this measure in terms of  $\epsilon$ -dominance. Therefore, we expect that this measure is easier to understand by the user than the measures used in the other sandwich algorithms.

## 6 Transformations

### 6.1 Notation for transformations

Up to now, we have assumed that all objective functions  $f_i$ ,  $i = 1, \dots, m$ , are convex. However in practice, there are many multi-objective optimization problems for which one or more objectives are not convex. For bi-criteria problems, Siem et al. (2008) have shown that, under certain conditions, it is possible to transform non-convex objective functions in such a way that an  $IPS$  and  $OPS$  can be determined for a non-convex set  $PS$ . Moreover, they show that if both objective functions are already convex, transforming the objective functions can result in better  $IPS$  and  $OPS$ .

In this section, we show that similar results can be obtained for multi-objective optimization problems with more than two objectives. To show these results, we introduce the following notation. The transformation function  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$  is defined as  $h([z_1, \dots, z_m]^\top) = [h_1(z_1), \dots, h_m(z_m)]^\top$ . The function  $h_i$  thus only transforms the output of the  $i^{\text{th}}$  objective function  $f_i$ . In other words, the vector function  $h$  consists of  $m$  functions  $h_i : \mathbb{R} \mapsto \mathbb{R}$ . Similarly, the inverse transformation function  $h^{-1} : \mathbb{R}^m \mapsto \mathbb{R}^m$  is a vector function consisting of the inverse functions of  $h_1, \dots, h_m$ . When discussing properties of these vector functions, we also consider the properties componentwise. For example, we call  $h$  convex if all  $h_i$ ,  $i = 1, \dots, m$ , are convex. We call the set of vectors  $\{h(z) \mid z \in Z\}$  the transformed objective space. To obtain the above mentioned results, we require that the transformation function  $h$  is strictly increasing. Therefore, we will assume for the remainder of this paper that  $h$  is strictly increasing. The corresponding inverse function  $h^{-1}$  is then also strictly increasing. Furthermore, to improve *IPS* and *OPS* for already convex *PS*, we require that  $h$  is strictly increasing and concave. This implies that  $h^{-1}$  exists, is strictly increasing and convex. We will mention explicitly when also concavity of  $h$  is required for a certain result.

To distinguish equivalent concepts in the transformed and original objective space, we use a different font to indicate concepts in the transformed space. This implies, for instance, that  $\mathcal{PS} := \{h(f(x)) \mid x \in X_E, h(f(x)) \leq h(z^{ub})\}$  where  $X_E$  is the set of solutions  $x$  for which there exists no  $\tilde{x} \in X$  such that  $h(f(\tilde{x})) \leq h(f(x))$ . The dummy points  $d^i(h(z))$  in the transformed space are defined as:

$$d_j^i(h(z)) = \begin{cases} h_j(z_j) & \text{if } j \neq i \\ mh_j(z_j^{ub}) + \theta & \text{if } j = i \end{cases} \quad \text{for } j = 1, \dots, m.$$

Furthermore, sets obtained by applying the inverse transformation  $h^{-1}$  to points in the transformed space are indicated by  $-1$  as a superscript. This implies that  $\text{conv}^{-1}\{\mathcal{IPS}, \mathcal{D}\}$  is defined as  $\{z \mid h(z) \in \text{conv}\{\mathcal{IPS}, \mathcal{D}\}\}$ . Similarly, applying  $h^{-1}$  to all points in  $\mathcal{H}(w, b) = \{h(z) \mid w^\top h(z) = b\}$  gives us  $\mathcal{H}^{-1}(w, b) = \{z \mid w^\top h(z) = b\}$  which we will call an inverted hyperplane. The set  $\mathcal{RH}^{-1}$  is defined as  $\{\mathcal{H}^{-1}(w, b) \mid \mathcal{H}(w, b) \in \mathcal{RH}\}$  where  $\mathcal{RH}$  is a set containing all relevant hyperplanes of  $\text{conv}\{\mathcal{IPS}, \mathcal{D}\}$ . Lastly, when  $\mathcal{F}$  is a facet of  $\text{conv}\{\mathcal{IPS}, \mathcal{D}\}$ , we refer to  $\mathcal{F}^{-1}$  as an inverted facet of  $\text{conv}^{-1}\{\mathcal{IPS}, \mathcal{D}\}$ .

## 6.2 Non-convex objectives

To obtain inner and outer approximations of *PS* when we have non-convex objectives, we can use the transformations in the following way. First, we have to find a transformation function  $h(z)$  which is strictly increasing in  $z$  and for which  $h(f(x))$  is convex in  $x$ . In order to use results proven in previous sections, we also assume that  $h(f(x)) \geq 0$  for all  $x \in X$ . This assumption is no real limitation as we can always satisfy this condition if the previous two conditions are met. If we have been able to find a transformation function  $h$  satisfying these properties, the transformed Pareto set  $\mathcal{PS}$  is convex. This implies that the sets *IPS* and *OPS* can be determined using one of the previously discussed sandwich algorithms. Because the transformation function is strictly increasing, the corresponding sets  $\mathcal{IPS}^{-1}$  and  $\mathcal{OPS}^{-1}$  are inner and outer approximations of  $\mathcal{PS}^{-1}$ . Finally, we can show that  $PS = \mathcal{PS}^{-1}$ , which implies that  $\mathcal{IPS}^{-1}$  and  $\mathcal{OPS}^{-1}$  are also inner and outer approximations of *PS*. We prove that the above method indeed produces inner and outer approximations of *PS* by the following proposition.

**Proposition 5.** *Let  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$  be a strictly increasing transformation function. If  $IPS$  and  $OPS$  are inner and outer approximations of  $PS$  then  $IPS^{-1}$  and  $OPS^{-1}$  are inner and outer approximations of  $PS$ . Furthermore,  $PS = PS^{-1}$ .*

*Proof.* As  $h$  is strictly increasing,  $h^{-1}$  exists and is also strictly increasing. By definition,  $IPS$  satisfies  $IPS \subseteq PS + \mathbb{R}_{\geq}^m$ . So for every  $h(z) \in IPS$  there exists a  $h(\hat{z}) \in PS$  such that  $h(\hat{z}) \leq h(z)$ . Because  $h^{-1}$  is strictly increasing, this condition implies that for every  $z \in IPS^{-1}$ , there exists a vector  $\hat{z} \in PS^{-1}$  such that  $\hat{z} \leq z$ , and so  $IPS^{-1} \subseteq PS^{-1} + \mathbb{R}_{\geq}^m$ . In the same way, we can also show that  $OPS^{-1}$  satisfies  $PS^{-1} \subseteq OPS^{-1} + \mathbb{R}_{\geq}^m$ . The sets  $IPS^{-1}$  and  $OPS^{-1}$  are thus inner and outer approximations of  $PS^{-1}$ .

To complete the proof, we now show that  $PS^{-1} = PS$ . Because  $h$  is strictly increasing,  $X_E$  is equal to  $X_E$  and  $PS$  is equal to  $\{h(z) \mid z \in PS\}$ . Using this formulation of  $PS$ , we can easily see that  $PS^{-1} = \{z \mid h(z) \in PS\}$  is equal to  $PS$ .  $\square$

The above results show how we can use  $IPS$  and  $OPS$  to find approximations  $IPS^{-1}$  and  $OPS^{-1}$  for  $PS$ . Similar to the untransformed case, we still need to determine which points of  $IPS^{-1}$  are non- $IPS^{-1}$ -dominated. As  $IPS^{-1}$  is in general not a convex hull of a finite number of points, we cannot directly use the method introduced in Section 4.3. The set  $IPS$ , on the other hand, does satisfy this condition. Therefore, we can use the method of Section 4.3 to determine the non- $IPS$ -dominated points of  $IPS$ . The following lemma shows that this also gives us the non- $IPS^{-1}$ -dominated points of  $IPS^{-1}$ .

**Lemma 7.** *Let  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$  be a strictly increasing transformation function. Then  $z \in IPS^{-1}$  is a non- $IPS^{-1}$ -dominated point if and only if the corresponding  $h(z) \in IPS$  is a non- $IPS$ -dominated.*

*Proof.* This result follows easily from the strict increasingness of  $h$ .  $\square$

### 6.3 Improving $IPS$ and $OPS$

To improve  $IPS$  and  $OPS$  when all objectives are convex, we also have to find a transformation function  $h$  satisfying several properties. Similar to the transformation function in the previous section,  $h(z)$  must be strictly increasing in  $z$  and  $h(f(x))$  must be convex in  $x$ . An additional property is that  $h$  must be a concave function. If we can determine a function  $h$  satisfying these properties, the following proposition shows us how we can improve  $IPS$ .

**Proposition 6.** *Let  $f : \mathbb{R}^m \mapsto \mathbb{R}^m$  be convex and  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$  be a strictly increasing and concave transformation function such that  $h \circ f$  is convex. Consider an inner approximation  $IPS = \text{conv}\{z^1, \dots, z^n\}$  of  $PS$ . If we take  $IPS = \text{conv}\{h(z) \mid z \in IPS\}$  and  $IPS^{-1} = \{z \mid h(z) \in IPS\}$ , then  $\alpha(z, IPS^{-1}) \leq \alpha(z, IPS)$  for every  $z \in PS$ .*

*Proof.* Because  $h$  is strictly increasing, we can easily see that the set  $\{h(z) \mid z \in IPS\}$  is an inner approximation of  $PS$ . Because  $h \circ f$  is convex,  $PS$  must be a convex set. Hence,  $IPS = \text{conv}\{h(z) \mid z \in IPS\}$  is also an inner approximation of  $PS$ . According to Proposition 5, the corresponding set  $IPS^{-1}$  must be an inner approximation of  $PS$ .

For a point  $z \in PS$ , let  $\hat{z} \in IPS$  be a point which  $\alpha$ -dominates  $z$  with  $\alpha = \alpha(z, IPS)$ . Because  $IPS = \text{conv}\{z^1, \dots, z^n\}$ ,  $\hat{z}$  can be written as  $\sum_{i=1}^n \lambda^i z^i$  with  $\sum_{i=1}^n \lambda^i = 1$  and  $\lambda^i \geq 0$  for  $i = 1, \dots, n$ . As  $h(z^i) \in IPS$  for all  $i = 1, \dots, n$ , the point  $\sum_{i=1}^n \lambda^i h(z^i)$  is an element of  $IPS$ . Subsequently, the point  $\tilde{z} = h^{-1}(\sum_{i=1}^n \lambda^i h(z^i))$  is an element of  $IPS^{-1}$ . Note that the



inverse function  $h^{-1}$  exists and is strictly increasing and convex because  $h$  is strictly increasing and concave. Using the convexity of  $h^{-1}$ , we can show the following:

$$\tilde{z} = h^{-1} \left( \sum_{i=1}^n \lambda^i h(z^i) \right) \leq \sum_{i=1}^n \lambda^i h^{-1}(h(z^i)) = \sum_{i=1}^n \lambda^i z^i = \hat{z}.$$

Because  $\tilde{z} \in \mathcal{IPS}^{-1}$  and  $\tilde{z} \leq \hat{z}$ , this proves that  $\alpha(z, \mathcal{IPS}^{-1}) \leq \alpha(z, \mathcal{IPS})$ .  $\square$

Note that when we take an  $\mathcal{IPS}$  obtained with one of the sandwich algorithms, the set  $\text{conv}\{\mathcal{IPS}, D\}$  also forms an inner approximation of  $PS$ . Therefore, the above lemma also holds when we replace  $\mathcal{IPS}$  by  $\text{conv}\{\mathcal{IPS}, D\}$ .

To show that also  $\mathcal{OPS}$  can be improved using this transformation, we first prove the following lemma which shows how we can find a supporting hyperplane of  $\mathcal{PS}$  when we have a supporting hyperplane of  $PS$ .

**Lemma 8.** *Let  $f : \mathbb{R}^m \mapsto \mathbb{R}^m$  and  $X$  be convex and  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$  be a strictly increasing and concave transformation function such that  $h \circ f$  is convex. Consider a hyperplane  $H(w, b)$  which supports  $PS$  at  $\bar{z} \in Z$ . Then hyperplane  $\mathcal{H}(\tilde{w}, \tilde{b})$  with  $\tilde{w}_i = w_i(h_i^{-1})'(\bar{y}_i)$  for  $i = 1, \dots, m$  and  $\tilde{b} = \tilde{w}^\top \bar{y}$  supports  $\mathcal{PS}$  at  $\bar{y} = h(\bar{z})$ .*

*Proof.* Let  $p(z)$  be a continuous non-decreasing function such that  $PS = \{z \mid p(z) = 0, z \in Z\}$ . Note that such a function exists because  $f$  and  $X$  are convex, which implies that  $PS$  is convex, non-decreasing and connected (see, e.g., Miettinen (1999)). As  $PS = \mathcal{PS}^{-1}$  according to Proposition 5, this expression for  $PS$  implies that  $\mathcal{PS} = \{y \mid p(h^{-1}(y)) = 0, y \in Y\}$  where  $Y := \{h(z) \mid z \in Z\}$ . As  $H(w, b)$  supports  $PS$  at  $\bar{z}$ , we know that  $w$  is a subgradient of  $p(z)$  at  $\bar{z}$ , i.e.,  $\frac{\partial p(\bar{z})}{\partial z} = w$ . Using the following chain rule for subgradients (cf. Theorem 10.6 of Rockafellar and Wets (1998)) at  $\bar{y} = h(\bar{z})$ :

$$\frac{\partial p(h^{-1}(\bar{y}))_i}{\partial y_i} = \frac{\partial p(\bar{z})}{\partial z_i} (h_i^{-1})'(\bar{y}_i) \text{ for } i = 1, \dots, m,$$

we obtain that  $\tilde{w}$  with  $\tilde{w}_i = w_i(h_i^{-1})'(\bar{y}_i)$  for  $i = 1, \dots, m$  is a subgradient of  $\mathcal{PS}$ . Therefore, we can conclude that  $\mathcal{H}(\tilde{w}, \tilde{b})$  is a supporting hyperplane of  $\mathcal{PS}$  at  $\bar{y}$ .  $\square$

In Proposition 7, we show that transforming back the supporting hyperplane of  $\mathcal{PS}$  obtained in Lemma 8 results in a better  $\mathcal{OPS}$ .

**Proposition 7.** *Let  $f : \mathbb{R}^m \mapsto \mathbb{R}^m$  and  $X$  be convex and  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$  be a strictly increasing and concave function such that  $h \circ f$  is convex. Consider a hyperplane  $H(w, b)$  which supports  $PS$  at  $\bar{z} \in Z$  and the corresponding hyperplane  $\mathcal{H}(\tilde{w}, \tilde{b})$  which supports  $\mathcal{PS}$  at  $\bar{y} = h(\bar{z})$ , with  $\tilde{w}_i = w_i(h_i^{-1})'(\bar{y}_i)$  for  $i = 1, \dots, m$  and  $\tilde{b} = \tilde{w}^\top \bar{y}$ . Then  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b})$  is a tighter outer approximation of  $PS$  than  $H(w, b)$ , i.e.,  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b})$  is an outer approximation of  $PS$  and  $H(w, b)$  is an outer approximation of  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b})$ .*

*Proof.* First we show that  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b})$  is an outer approximation of  $PS$ . Since  $\mathcal{H}(\tilde{w}, \tilde{b})$  supports  $\mathcal{PS}$  at  $\bar{y}$ , it follows that:

$$\forall y^1 \in \mathcal{PS}, \exists y^2 \in \mathcal{H}(\tilde{w}, \tilde{b}) \quad : \quad y^2 \leq y^1.$$

Because  $h^{-1}$  is strictly increasing and  $PS = \mathcal{PS}^{-1}$ , this is equivalent to:

$$\forall z^1 \in PS, \exists z^2 \in \mathcal{H}^{-1}(\tilde{w}, \tilde{b}) \quad : \quad z^2 \leq z^1.$$

Hence  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b})$  is an outer approximation of  $PS$ .

Now we prove that  $H(w, b)$  is an outer approximation of  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b})$ . Let  $p(z)$  be a non-decreasing function such that  $PS = \{z \mid p(z) = 0, z \in Z\}$ . This implies that  $\mathcal{PS} = \{y \mid p(h^{-1}(y)) = 0, y \in Y\}$  where  $Y := \{h(z) \mid z \in Z\}$ . The supporting hyperplane  $\mathcal{H}(\tilde{w}, \tilde{b})$  at  $\bar{y} = h(\bar{z})$  of Lemma 8 can now be written as

$$\mathcal{H}(\tilde{w}, \tilde{b}) = \left\{ y \mid \sum_i w_i (h_i^{-1})'(\bar{y}_i) [y_i - \bar{y}_i] = 0 \right\}, \quad (10)$$

in which  $w = \frac{\partial p(\bar{z})}{\partial z}$  is a subgradient of  $p(z)$  at  $\bar{z}$ . Transforming  $\mathcal{H}(\tilde{w}, \tilde{b})$  back, we get  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b}) = \{z \mid p_2(z) = 0\}$ , where

$$p_2(z) = \sum_i w_i (h_i^{-1})'(\bar{y}_i) [h_i(z_i) - h_i(\bar{z}_i)].$$

Moreover, we have  $H(w, b) = \{z \mid p_1(z) = 0\}$ , where

$$p_1(z) = \sum_i w_i [z_i - \bar{z}_i].$$

For the  $i^{th}$  term of  $p_2(z)$ , we have:

$$(h_i^{-1})'(\bar{y}_i) [h_i(z_i) - h_i(\bar{z}_i)] \leq (h_i^{-1})'(\bar{y}_i) [h_i(\bar{z}_i) + h_i'(\bar{z}_i)(z_i - \bar{z}_i) - h_i(\bar{z}_i)] = z_i - \bar{z}_i,$$

where the inequality follows because  $h_i$  is concave and the equality follows because  $(h_i^{-1})' = \frac{1}{h_i'}$  and  $h_i^{-1}$  is strictly increasing. Since  $p(z)$  is non-decreasing and thus  $\frac{\partial p(\bar{z})}{\partial z_i} \geq 0$ , we have  $p_2(z) \leq p_1(z)$ . We thus have  $p_1(z) \geq 0, \forall z \in \mathcal{H}^{-1}(\tilde{w}, \tilde{b})$ , which means that  $H(w, b)$  is an outer approximation of  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b})$ .  $\square$

#### 6.4 Calculating $\alpha(PS, IPS^{-1})$ and $\alpha(OP\mathcal{S}^{-1}, IPS^{-1})$

The general idea behind the method for calculating  $\alpha(PS, IPS^{-1})$  and  $\alpha(OP\mathcal{S}^{-1}, IPS^{-1})$  is the same as for  $\alpha(PS, IPS)$  and  $\alpha(OP\mathcal{S}, IPS)$  as described in Section 5.2. The main difference is that in all steps  $H(w, b)$  must be replaced by  $\mathcal{H}^{-1}(w, b)$ . This has the drawback that  $\max_{z \in OP\mathcal{S}^{-1}} \alpha(z, \mathcal{H}^{-1}(w, b))$  can no longer be calculated as an LP-problem. Instead, we have to solve a non-convex problem. To prove that all steps made in the algorithm still hold, we first prove the following lemma which is similar to Lemma 4.

**Lemma 9.** *Let  $\epsilon \in \mathbb{R}_{>}^m$ ,  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$  be a strictly increasing and continuous transformation function and  $\mathcal{H}^{-1}(w, b) \in \mathcal{RH}^{-1}$ . For every  $z \in PS$ ,  $\alpha(z, \mathcal{H}^{-1}(w, b))$  is equal to the unique  $\hat{\alpha}$  for which the point  $z + \hat{\alpha}\epsilon$  is on  $\mathcal{H}^{-1}(w, b)$ .*

*Proof.* Let  $\hat{z}$  be an arbitrary point on  $\mathcal{H}^{-1}(w, b)$ . As  $\epsilon \in \mathbb{R}_{>}^m$ , we can always determine an  $\alpha$  such that  $z + \alpha\epsilon > \hat{z}$ . Because  $h$  is strictly increasing and  $w \geq 0$ ,  $w^\top h(z + \alpha\epsilon) > w^\top h(\hat{z}) = b$ . Similarly, we can prove that there must exist an  $\alpha$  such that  $w^\top h(z + \alpha\epsilon) < b$ . As  $h$  is a strictly

increasing continuous function, there must thus exist a unique  $\hat{\alpha}$  such that  $w^\top h(z + \hat{\alpha}\epsilon) = b$ . As  $z$  is  $\alpha\epsilon$ -dominated by  $z + \hat{\alpha}\epsilon \in \mathcal{H}^{-1}(w, b)$ , we know that  $\alpha(z, \mathcal{H}^{-1}(w, b)) \leq \hat{\alpha}$ . If  $\alpha(z, \mathcal{H}^{-1}(w, b)) < \hat{\alpha}$ , there must exist a  $\tilde{z} \in \mathcal{H}^{-1}(w, b)$  such that  $\tilde{z} < z + \hat{\alpha}\epsilon$ . However, this would imply:

$$b = w^\top h(\tilde{z}) < w^\top h(z + \hat{\alpha}\epsilon) = b.$$

Due to this contradiction, we can conclude that  $\alpha(z, \mathcal{H}^{-1}(w, b)) = \hat{\alpha}$ .  $\square$

Using the results of Lemma 1, 2 and 9, we can prove the following two lemmas.

**Lemma 10.** *Let  $\epsilon \in \mathbb{R}_{>}^m$  and  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$  be a strictly increasing transformation function. Consider  $z \in PS$  and  $\mathcal{H}^{-1}(w, b) \in \mathcal{RH}^{-1}$ . Then  $\alpha(z, IPS^{-1}) \geq \alpha(z, \mathcal{H}^{-1}(w, b))$ .*

*Proof.* According to Lemma 2,  $w \geq 0$ . Assume by contradiction that  $z$  is  $\hat{\alpha}\epsilon$ -dominated by  $IPS^{-1}$  with  $\hat{\alpha} < \alpha(z, \mathcal{H}^{-1}(w, b))$ . This implies that there should exist a point  $\tilde{z} \in IPS^{-1}$  such that  $\tilde{z} \leq z + \hat{\alpha}\epsilon$ . For this point  $\tilde{z}$  it holds:

$$w^\top h(\tilde{z}) \leq w^\top h(z + \hat{\alpha}\epsilon) < w^\top h(z + \alpha(z, \mathcal{H}^{-1}(w, b))\epsilon) = b,$$

where the last equality follows from Lemma 9. As  $\mathcal{H}^{-1}(w, b) \in \mathcal{RH}^{-1}$ , for all points  $z$  in  $IPS^{-1}$  it must hold that  $w^\top h(z) \geq b$ . This implies that  $\tilde{z}$  cannot be an element of  $IPS^{-1}$ . Therefore,  $z$  can only be  $\alpha\epsilon$ -dominated by  $IPS^{-1}$  with  $\alpha \geq \alpha(z, \mathcal{H}^{-1}(w, b))$ .  $\square$

**Lemma 11.** *Consider  $\epsilon \in \mathbb{R}_{>}^m$ , a strictly increasing transformation function  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$  and  $z \in PS$ . Then  $\alpha(z, IPS^{-1}) = \alpha(z, \mathcal{H}^{-1}(w, b))$  must hold for at least one  $\mathcal{H}^{-1}(w, b) \in \mathcal{RH}^{-1}$ .*

*Proof.* Define  $z^\alpha = z + \alpha(z, IPS^{-1})\epsilon$  and let  $z^{IPS^{-1}} \in IPS^{-1}$  be such that  $\alpha(z, \{z^{IPS^{-1}}\}) = \alpha(z, IPS^{-1})$ . By the definition of  $\alpha(z, \cdot)$ , we know that  $z^{IPS^{-1}} \leq z^\alpha$  with equality for at least one coordinate. Furthermore, there cannot exist a point  $\hat{z} \in IPS^{-1}$  such that  $\hat{z} < z^{IPS^{-1}}$  or  $\hat{z} < z^\alpha$ . Because all points in  $\mathcal{D}$  are (weakly) dominated by points in  $IPS$  and  $h^{-1}(z)$  is strictly increasing, the same is true for all points  $\hat{z}$  in  $\text{conv}^{-1}\{IPS, \mathcal{D}\}$ . This implies that  $z^{IPS^{-1}}$  must lie on one or more inverted facets of  $\text{conv}^{-1}\{IPS, \mathcal{D}\}$ . Let  $\mathcal{F}^{-1}$  be one of these inverted facets and  $\mathcal{F}$  the associated facet in the transformed space. Because  $h(z^{IPS^{-1}}) \leq h(z^{ub})$  and  $h(z^{IPS^{-1}}) \in \mathcal{F}$ ,  $\mathcal{F}$  must be a relevant facet of  $\text{conv}\{IPS, \mathcal{D}\}$  according to Lemma 1. Consequently,  $\mathcal{F}^{-1}$  is a relevant inverted facet which has a supporting inverted hyperplane  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b}) \in \mathcal{RH}^{-1}$ . For this inverted hyperplane it holds that  $z^{IPS^{-1}} \in \mathcal{F}^{-1} \subset \mathcal{H}^{-1}(\tilde{w}, \tilde{b})$ . Using this information, we can show the following relation:

$$\alpha(z, \mathcal{H}^{-1}(\tilde{w}, \tilde{b})) \leq \alpha(z, \{z^{IPS^{-1}}\}) = \alpha(z, IPS^{-1}).$$

The inequality follows from the definition of  $\alpha(z, \cdot)$  and the fact that  $z^{IPS^{-1}} \in \mathcal{H}^{-1}(\tilde{w}, \tilde{b})$ . The equality follows from the definition of  $z^{IPS^{-1}}$ . Combining the above equation with Lemma 10 gives that  $\alpha(z, IPS^{-1}) = \alpha(z, \mathcal{H}^{-1}(\tilde{w}, \tilde{b}))$  must hold for this  $\mathcal{H}^{-1}(\tilde{w}, \tilde{b}) \in \mathcal{RH}^{-1}$ .  $\square$

By combining Lemma 10 and 11, we obtain the following equation:

$$\alpha(z, IPS^{-1}) = \max_{\mathcal{H}^{-1}(w, b) \in \mathcal{RH}^{-1}} \alpha(z, \mathcal{H}^{-1}(w, b)), \quad (11)$$

which gives an expression for  $\alpha(z, IPS^{-1})$ . By definition, to determine  $\alpha(PS, IPS^{-1})$ , we must take the maximum of  $\alpha(z, IPS^{-1})$  over all  $z \in PS$ . However determining  $\alpha(z, IPS^{-1})$  explicitly for every  $z \in PS$  is not an option as there are an infinite number of vectors in the set  $PS$ . The number of  $\mathcal{H}^{-1}(w, b) \in \mathcal{RH}^{-1}$ , on the other hand, is finite. Therefore, we decide to calculate  $\max_{z \in PS} \alpha(z, \mathcal{H}^{-1}(w, b))$  for every  $\mathcal{H}^{-1}(w, b) \in \mathcal{RH}^{-1}$  and use these results to calculate  $\alpha(PS, IPS^{-1})$ . The following lemma gives a method to determine  $\max_{z \in PS} \alpha(z, \mathcal{H}^{-1}(w, b))$  without knowing  $PS$  explicitly.

**Lemma 12.** *Consider a hyperplane  $\mathcal{H}^{-1}(w, b) \in \mathcal{RH}^{-1}$ , a strictly increasing transformation function  $h : \mathbb{R}^m \mapsto \mathbb{R}^m$ , and  $\epsilon \in \mathbb{R}_{>0}^m$ . Then  $\max_{z \in PS} \alpha(z, \mathcal{H}^{-1}(w, b))$  can be determined by solving the following optimization problem:*

$$\begin{aligned} \max_{\alpha, z} \quad & \alpha \\ \text{s.t.} \quad & z \in Z \\ & w^\top h(z + \alpha\epsilon) \leq b. \end{aligned} \tag{12}$$

*Proof.* Let us define  $g(z, \alpha) = w^\top h(z + \alpha\epsilon)$ . As  $w \geq 0$ ,  $\epsilon > 0$  and  $h$  is strictly increasing,  $g$  is a strictly increasing function in  $\alpha$ . Because we maximize  $\alpha$ , this implies that we can replace the last inequality constraint by the equality constraint  $w^\top h(z + \alpha\epsilon) = b$ .

We now show that there always exists a solution  $\hat{z} \in PS$  for this problem. Let us denote an optimal  $z$  and  $\alpha$  of the above problem by  $z^*$  and  $\alpha^*$ . We can easily show that there cannot exist an  $z \in Z$  such that  $z < z^*$  as this would imply that  $\alpha^*$  is not optimal. Therefore, either  $z^*$  is in  $PS$  or  $z^*$  is weakly dominated by another point  $\hat{z} \in PS$ . In the first case, we can simply take  $\hat{z} = z^*$ . In the second case, because  $w \geq 0$  and  $h$  strictly increasing, we can show the following:

$$w^\top h(\hat{z} + \alpha^*\epsilon) \leq w^\top h(z^* + \alpha^*\epsilon) = b.$$

When  $\hat{\alpha}$  satisfies  $w^\top h(\hat{z} + \hat{\alpha}\epsilon) = b$ , then  $\hat{\alpha} \geq \alpha^*$  because  $g(z, \alpha)$  is strictly increasing in  $\alpha$ . However,  $\hat{\alpha} > \alpha^*$  cannot hold because  $\alpha^*$  is optimal. We can thus conclude that  $\hat{\alpha} = \alpha^*$  and that  $\hat{z} \in PS$  is also an optimal solution. The constraint  $z \in Z$  can thus be replaced by  $z \in PS$ . We know from Lemma 9 that for every  $z \in PS$ ,  $\alpha(z, \mathcal{H}^{-1}(w, b))$  is equal to the unique  $\alpha$  which satisfies the equality constraint  $w^\top h(z + \alpha\epsilon) = b$ . This proves that we can determine  $\max_{z \in PS} \alpha(z, \mathcal{H}^{-1}(w, b))$  by solving the above problem.  $\square$

By solving the above problem for every  $\mathcal{H}(w, b) \in \mathcal{RH}$ , we can thus determine the value of  $\max_{z \in PS} \alpha(z, \mathcal{H}(w, b))$  explicitly for every  $\mathcal{H}(w, b) \in \mathcal{RH}$ . Combining the above results, we can use the following equation to calculate  $\alpha(PS, IPS^{-1})$ :

$$\begin{aligned} \alpha(PS, IPS^{-1}) &= \max_{z \in PS} \alpha(z, IPS^{-1}) = \max_{z \in PS} \max_{\mathcal{H}(w, b) \in \mathcal{RH}} \alpha(z, \mathcal{H}(w, b)) \\ &= \max_{\mathcal{H}(w, b) \in \mathcal{RH}} \max_{z \in PS} \alpha(z, \mathcal{H}(w, b)). \end{aligned}$$

## 7 Application of enhancements

### 7.1 Application of dummy points

In previous sections, we have seen that adding dummy points to  $IPS$  has a number of benefits. These benefits are not just for one specific sandwich algorithm but apply to various sandwich

algorithms. Some benefits even apply to general MOP-methods for approximating a convex set  $PS$ . To illustrate this, we will describe various possible applications of dummy points.

Adding dummy points to  $IPS$  to solve the problem of 'undesirable' normals can be useful for any MOP-algorithm which uses normals of facets as weights to determine Pareto points. The algorithms of Solanki et al. (1993) and Craft et al. (2006) are just two examples of such algorithms. By eliminating the relevant facets with 'undesirable' normals, no special rules have to be constructed to deal with these facets.

The dummy points can be used to determine  $IPS^{nId}$  when  $IPS$  is a convex hull of a finite number of points. Besides this condition on the form of  $IPS$ , no other conditions have to be satisfied in order to apply the dummy points for this purpose. After adding the dummy points to  $IPS$ , we can simply use the algorithm described in Section 4.3 to determine all non- $IPS$ -dominated points.

## 7.2 Application of error measure

Calculating  $\alpha(PS, IPS)$  by using  $\text{conv}\{IPS, D\}$  can be done for any  $IPS$  which is the convex hull of a finite number of points. The calculation of  $\alpha(PS, IPS)$  is mainly useful if we want to know the real approximation error of  $IPS$ . However, in real life applications, using the method in Section 5 to calculate  $\alpha(PS, IPS)$  is generally too time consuming as it requires solving a large number of weighted sum problems. Furthermore, the outcomes of these weighted sum problems could be used to improve  $IPS$  and  $OPS$ . After this improvement, the calculated value of  $\alpha(PS, IPS)$  is probably not the real accuracy of the new  $IPS$ , but just an upper bound.

The calculation of  $\alpha(OPS, IPS)$  does not have the above problems. As the weighted sum problems become LP-problems, calculating  $\alpha(OPS, IPS)$  can be done much faster. Another difference is that the outcomes of these optimizations cannot be used to improve  $IPS$ . Because the measure  $\alpha(OPS, IPS)$  obviously requires an  $IPS$  and  $OPS$ , it can only be calculated for sandwich-algorithms. Furthermore, we can use the measure also within a sandwich algorithm to decide which facet to use for a next optimization. In Section 7.4, we describe how this can be done in the algorithm of Solanki et al. (1993).

## 7.3 Application of transformations

The transformations described in Section 6 can be divided into two cases. Firstly, we can use transformations to determine  $OPS$  and  $IPS$  if we have non-convex objectives. In this case, we must determine for each non-convex objective  $f_i$  a transformation function  $h_i$  which is strictly increasing and for which  $h_i \circ f_i$  is convex. When this is possible, we can use any of the sandwich algorithms to determine  $IPS$  and  $OPS$  which approximate  $PS$ . These approximations can subsequently be used to determine  $IPS^{-1}$  and  $OPS^{-1}$  as approximations of  $PS$ .

Secondly, we can transform objectives to obtain tighter  $OPS$  and  $IPS$  for a convex  $PS$ . To do this, we must find a strictly increasing and concave function  $h$  such that  $h \circ f$  is convex. We can use this transformation to improve the final  $IPS$  and  $OPS$  obtained with any of the sandwich algorithms. Furthermore, instead of using  $\alpha(OPS, IPS)$  to select a next facet, we also can use  $\alpha(OPS^{-1}, IPS^{-1})$ . However, a drawback of using  $\alpha(OPS^{-1}, IPS^{-1})$  is that it is more time-consuming to calculate than  $\alpha(OPS, IPS)$ . We thus have to decide whether the additional calculation time is justified by a possibly better choice of facets.

Objectives to which we can apply the above transformations can, among others, be found in geometric programming and IMRT optimization. An example in geometric programming is treated in Section 8.5. For transformations of IMRT objectives, we refer to the paper of Hoffmann et al. (2008).

## 7.4 Enhanced version of algorithm of Solanki et al.

All above described enhancements are aimed at determining an accurate approximation of a convex (or non-convex)  $PS$  more efficiently. To test whether these enhancements can indeed improve efficiency, we enhance the algorithm of Solanki et al. (1993) by adding dummy points and changing the error measure. We do not include the use of transformations as this is not possible for all (convex) MOPs. More specifically, the enhanced version of the algorithm of Solanki et al. (1993) differs from the original version in the following ways.

In every step of the algorithm,  $IPS$  is replaced by  $\text{conv}\{IPS, D\}$ . As shown in Section 5.2, this enables us to easily calculate  $\alpha(OPS, IPS)$  by solving Problem 3 for all relevant facets. According to the result of Proposition 4, we must use  $\frac{b-w^\top \bar{z}}{w^\top \epsilon}$  instead of  $b - w^\top \bar{z}$  as the error of a facet where  $\bar{z}$  is the solution of Problem 3. By calculating the error of the facets in this way,  $\alpha(OPS, IPS)$  is equal to the maximum of the errors of all relevant facets.

Although the algorithms of Craft et al. (2006) and Klamroth et al. (2002) can also be enhanced, we have not tested the enhanced versions of these algorithms. If we would enhance the algorithm of Craft et al. (2006) with the use of dummy points, the algorithm would become very similar to the enhanced version of the algorithm of Solanki et al. (1993). The only difference that would remain is the difference in the error used to select the facets. Enhancing the algorithm of Craft et al. (2006) with both the dummy points and the error measure would make the algorithms completely the same. The algorithms of Klamroth et al. (2002) were not enhanced because initial tests showed that these algorithms are much less efficient than the algorithms of Solanki et al. (1993) and Craft et al. (2006). A main reason for this is that not all obtained Pareto points are added to  $IPS$ . As this drawback is not reduced by the enhancements, it seems unlikely that an enhanced version of the algorithm of Klamroth et al. (2002) would become more efficient than the other sandwich algorithms.

# 8 Numerical comparison of different sandwich algorithms

## 8.1 Comparison method

In this numerical comparison, we consider the following five algorithms:

- Algorithm of Solanki et al. (1993) (SOLANKI).
- Algorithm of Klamroth et al. (2002) with equality constraint (KLAMROTH=).
- Algorithm of Klamroth et al. (2002) with inequality constraint (KLAMROTH $\leq$ ).
- Algorithm of Craft et al. (2006) (CRAFT).
- Enhanced version of algorithm of Solanki et al. (1993) (ENHANCED).

To compare these algorithms, we use three different test cases. The first two cases are artificial test cases and the third case is an example of an IMRT problem. To show the effect

of transforming convex objective functions, a fourth case is used which consists of a geometric programming problem. A more elaborate description of these four cases is given in Sections 8.2, 8.3, 8.4, and 8.5.

For all cases, we must set a number of parameters. The parameter  $z^{ub}$  is taken equal to the pseudo-nadir point  $z^{pN}$  as defined in Section 2. To calculate the error measure, we must also set the vector  $\epsilon$  which specifies for each objective a maximal allowable error. As the objectives in cases 1 and 2 have no real interpretation, it is not immediately clear how to select values for  $\epsilon$ . A reasonable choice could be to take  $\epsilon$  equal to the difference between the vectors  $z^U$  and  $z^{ub}$  which contain the minimal and maximal possible values of all objectives. By setting  $\epsilon$  in this way, we try to give each objective equal importance. However, we are not interested in an  $\epsilon$ -approximation for this particular  $\epsilon$ , because any point  $z \in Z$  forms an  $\epsilon$ -approximation of  $PS$ . Instead, we are interested in reaching a certain value for  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$ . For example, if we want a maximal error of 5 percent of the difference between the minimal and maximal objective values, then we want to determine when  $\alpha(PS, IPS) \leq 0.05$  or  $\alpha(OPS, IPS) \leq 0.05$  is reached.

To measure the efficiency, we assume that solving an optimization problem to find a Pareto point is a difficult problem and relatively the most time intensive part of each of the approximation algorithms. Because the optimizations in the first two test cases are quite easy to solve, the CPU-time used by the algorithms is not representative for real-life cases. Therefore, we measure the computational effort of the algorithms by the number of performed optimizations. In CRAFT, SOLANKI and ENHANCED, we thus measure the number of weighted sum problems. In KLAMROTH $^{\leq}$  and KLAMROTH $^=$ , we measure the number of runs of Problems 4 and 5. For each algorithm, we denote the total number of optimizations performed at a certain stage of the algorithm by  $n^{opt}$ . Although the anchor-points in each algorithm also have to be determined by solving optimization problems, these optimizations are not included in  $n^{opt}$  as they are often easier to solve and because they are the same for all algorithms.

As mentioned in the description of KLAMROTH $^{\leq}$  and KLAMROTH $^=$  in Section 3.3, not all points and inequalities determined by Problems 4 and 5 are used to improve  $IPS$  and  $OPS$ . This does not seem very efficient as determining these points and inequalities is time-consuming. Therefore, we also determine the inner and outer approximations which use all available points and inequalities and denote these approximations by  $IPS^*$  and  $OPS^*$ .

Lastly, using only  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$  to measure the accuracy seems to give an unfair advantage to ENHANCED as this algorithm uses  $\alpha(OPS, IPS)$  to select a facet in each iteration. Therefore, we also calculated the accuracy of the different  $IPS$  and  $OPS$  using the error measures used in SOLANKI and CRAFT. However, the measure used in SOLANKI also calculates the distance between  $IPS$  and  $OPS$  for facets with an 'undesirable' normal. The maximum of all distances was often attained by one of these facets. A large distance for one of these facets not necessarily implies a large inaccuracy near these facets. Especially for  $IPS$ s obtained by CRAFT, the measure of SOLANKI often returned the same value for  $IPS$ s with significantly different values for  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$ . Due to this drawback, we decided not to use this measure. The measure used in CRAFT has a different drawback. When adding a new point to  $IPS$  to obtain a new inner approximation  $IPS^*$ , the measure could give a higher value for  $IPS^*$  than for  $IPS$ . As an improvement of  $IPS$  could result in a deterioration of this error measure, we also decided not to use this measure in our comparison.

## 8.2 Test case 1

The first test case has the following three objectives and constraints:

$$\begin{aligned}
f_1(x) &= x_1 \\
f_2(x) &= x_2 \\
f_3(x) &= x_3 \\
x_1 &\geq (x_2 - 9)^2 + (x_3 - 3)^2 \\
x_2 &\geq (x_1 - 4)^2 + (x_3 - 3)^2 \\
x_3 &\geq (x_1 - 4)^2 + (x_2 - 9)^2.
\end{aligned}$$

It is easy to see that all objectives and the feasible region are convex.

We compare the *IPS* and *OPS* of the different algorithms on both  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$ . The first measure is important because it shows the real accuracy of *IPS*. However in practice, we will generally only have  $\alpha(OPS, IPS)$  as calculating  $\alpha(PS, IPS)$  is too time-consuming. The decision maker will thus use  $\alpha(OPS, IPS)$  to determine if a certain approximation is accurate enough.

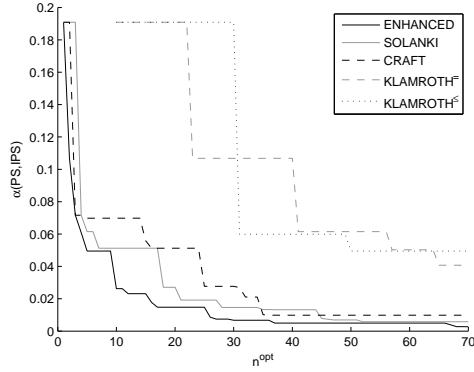


Figure 5:  $\alpha(PS, IPS)$  of test case 1 as function of  $n^{opt}$ .

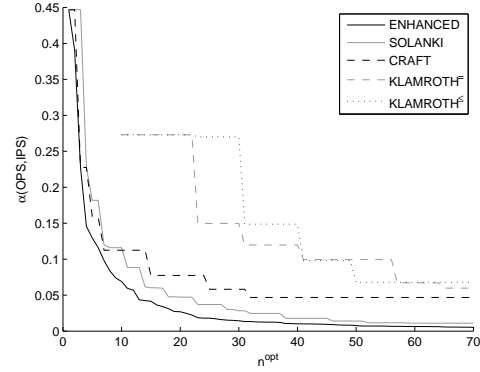


Figure 6:  $\alpha(OPS, IPS)$  of test case 1 as function of  $n^{opt}$ .

In Figures 5 and 6, we show for different values of  $n^{opt}$  the achieved accuracy of the five algorithms. As mentioned before, the inefficiency of adding only a selection of the obtained points and inequalities to *IPS* and *OPS*, make  $KLAMROTH^{\leq}$  and  $KLAMROTH^{=}$  the least efficient. When considering  $IPS^*$  and  $OPS^*$ , the results improve but remain considerably worse than the other algorithms. Therefore, we did not plot  $\alpha(PS, IPS^*)$  and  $\alpha(OPS^*, IPS^*)$  in the above figures. The algorithms CRAFT, SOLANKI and ENHANCED perform considerably better. For most values of  $n^{opt}$ , ENHANCED performs the best of the five tested algorithms.



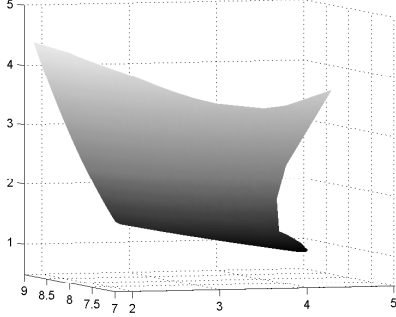


Figure 7:  $PS$  of test case 1.

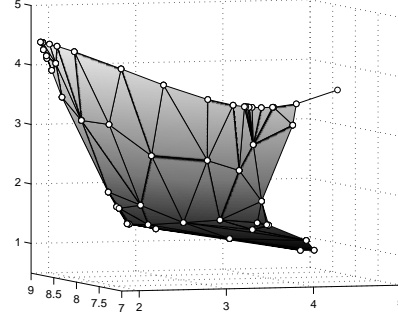


Figure 8: Approximation of CRAFT.

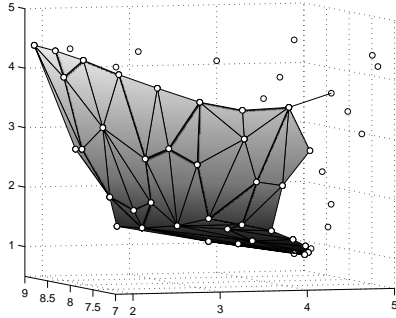


Figure 9: Approximation of SOLANKI.

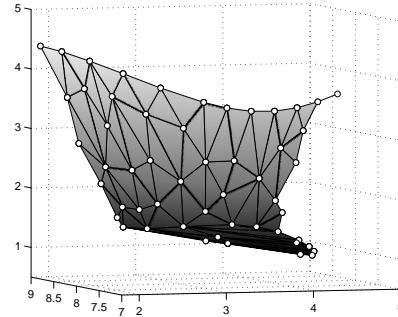


Figure 10: Approximation of ENHANCED.

As this MOP has three objectives, it is possible to draw  $PS$  and compare the different results graphically. In Figure 7, we show the set  $PS$  of this problem. In Figures 8, 9 and 10, the lines and shaded area show  $IPS^{nId}$  obtained when  $n^{opt} = 50$ . The dots represent the anchor points and all points found by the optimizations in the sandwich algorithms. When looking at the approximation obtained with CRAFT in Figure 8, we notice that there are some clusters of points. These may be caused by the method used in CRAFT to select a next weighting vector for the weighted sum method. The approximations of SOLANKI and ENHANCED also appear to contain a cluster of points in the lower left part of the figure. These points, however, only appear to be clustered due to the angle at which we view the approximation and are thus not really clustered. Although the method SOLANKI generates no clusters, it does have another drawback as we can see in Figure 9. Of the 50 points found by the optimizations 14 are  $IPS$ -dominated and thus no element of  $IPS^{nId}$ . All these points are obtained by using the weighted sum with an 'undesirable' normal. Figure 10 shows us that the  $IPS$  obtained with ENHANCED does not have these drawbacks as all points are nicely distributed and are part of  $IPS^{nId}$ .

### 8.3 Test case 2

To test the performance for MOPs with more dimensions, we use a linear five-dimensional MOP as the second test case. Similar to the first case, we set the objective functions equal to

the variables. To determine the constraints, 30 points were randomly drawn from a uniform distribution on the interval  $[0, 1]^5$ . The convex hull of these points was taken as the feasible region  $X$ .

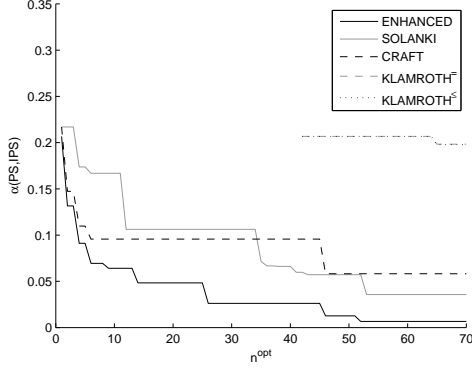


Figure 11:  $\alpha(PS, IPS)$  of test case 2 as function of  $n^{opt}$ .

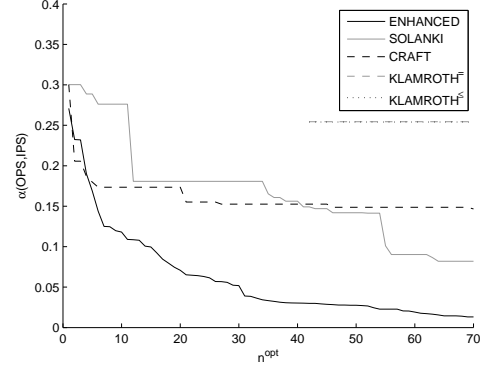


Figure 12:  $\alpha(OPS, IPS)$  of test case 2 as function of  $n^{opt}$ .

In Figures 11 and 12, we plot the values of  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$  for different values of  $n^{opt}$ . Again  $KLAMROTH^{\leq}$  and  $KLAMROTH^{=}$  are the least efficient algorithms. CRAFT initially performs better than SOLANKI, but for  $n^{opt}$  above 34 and 45 SOLANKI performs better on  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$ , respectively. However for almost all values of  $n^{opt}$ , ENHANCED performs significantly better than the four other algorithms. If, for instance, we want a quality guarantee of  $\alpha(OPS, IPS) \leq 0.1$ , ENHANCED needs only 15 optimizations to obtain this level of accuracy. SOLANKI, on the other hand, needs 56 optimizations and the other three algorithms need more than 70 optimizations. These figures show that the enhancements can improve the efficiency significantly. Furthermore, Figure 11 shows us that not only the upper bound  $\alpha(OPS, IPS)$  found by ENHANCED is better but also the real accuracy  $\alpha(PS, IPS)$ . Although a better  $\alpha(OPS, IPS)$  generally implies a better  $\alpha(PS, IPS)$ , this is not always true as we can see when comparing the values of  $\alpha(OPS, IPS)$  and  $\alpha(PS, IPS)$  at  $n^{opt} = 40$  for SOLANKI and CRAFT.

#### 8.4 Test case 3

As IMRT is one of the common application areas, we also include a test case from this field to compare the algorithms. The IMRT optimization problem used in this comparison is a 2D phantom pancreas case. Figure 13 shows the tumor and the five nearby organs. The high energy photon beams used in radiation therapy to treat cancer tumors have to pass through surrounding tissue to reach the tumor. To reduce the risk of damaging healthy tissue, the radiation dose delivered to this tissue should be minimized. The five organs indicated in Figure 13 are especially sensitive to radiation and are therefore referred to as the organs-at-risk (OARs). The radiation dose delivered to these OARs should be limited while enough radiation should be delivered to the tumor to eradicate it. To calculate these doses, the relevant part of the body of the patient is discretized by dividing it into voxels. Using a dose influence matrix, the radiation dose delivered to each voxel can be calculated for a specific radiation plan. The objectives are often formulated in terms of the mean or maximum dose delivered to all voxels belonging to a tumor or OAR.

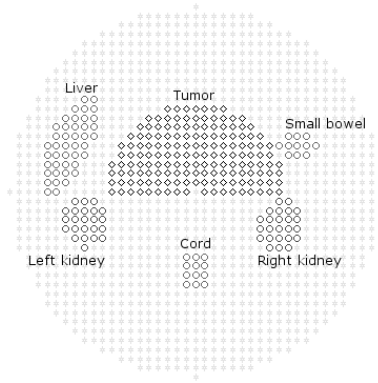


Figure 13: 2D pancreatic phantom case.

	Dose in Gy	
	Min.	Max.
Tumor	50	75
Left kidney		70
Right kidney		70
Spinal cord		45
Small bowel		75
Other tissue		75

Table 1: Minimal and maximal dose allowed for each structure.

In our problem, we consider four objectives. The first objective is aimed at delivering a dose of 60 Gray (Gy) to the tumor. Any voxel of the tumor which receives less than 60 Gy is penalized and the mean of these penalties forms the first objective. The three other objectives measure the maximum dose delivered to any of the voxels belonging to the left kidney, right kidney or spinal cord, respectively. Furthermore, constraints are added to ensure a minimal or maximal dose for certain structures. The values for these bounds can be found in Table 1. As all constraint and objectives are linear, this IMRT-problem is a multi-objective LP-problem and thus convex.

Figures 14 and 15 again contain the values of  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$ . KLAMROTH $^{\leq}$  and KLAMROTH $^=$  are not used in this test case as they performed significantly worse than the other algorithms in the previous two test cases. For this IMRT-case, CRAFT performs better than SOLANKI for all  $n^{opt}$ , but ENHANCED still performs better than both other algorithms. The quality guarantee of  $\alpha(OPS, IPS) \leq 0.1$  is reached by CRAFT after 42 optimizations but ENHANCED only needs 14. After 24 optimizations, ENHANCED even gives a quality guarantee of  $\alpha(OPS, IPS) \leq 0.05$ .

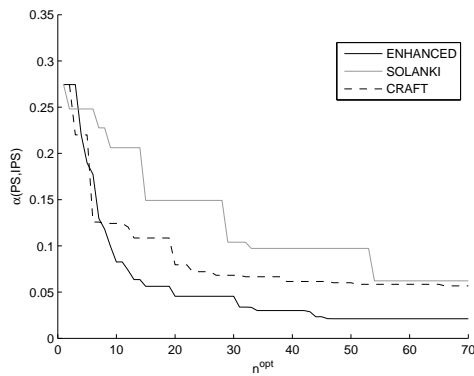


Figure 14:  $\alpha(PS, IPS)$  of test case 3 as function of  $n^{opt}$ .

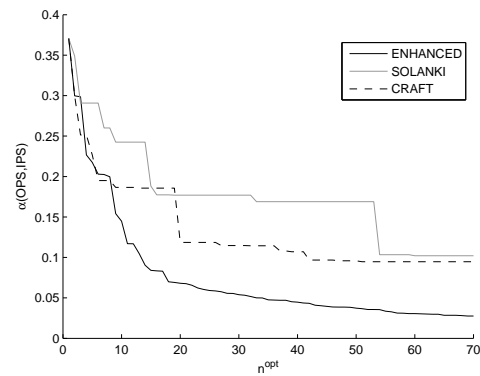


Figure 15:  $\alpha(OPS, IPS)$  of test case 3 as function of  $n^{opt}$ .

## 8.5 Test case 4

In the fourth and final test case, we consider the effect of transforming convex objectives. We test this using a geometric programming example. For a general introduction to geometric programming, we refer to Boyd et al. (2007).

In this test case, we consider the following geometric programming problem:

$$\begin{aligned}
f_1(x) &= e^{-x_1-x_2-x_3} \\
f_2(x) &= e^{x_4} \\
f_3(x) &= e^{x_5} \\
2e^{-x_4+x_1} (e^{x_2} + e^{x_3}) &\leq 1 \\
e^{-x_5+x_2+x_3} &\leq 1 \\
e^{|x(2)-x(1)|} &\leq 2 \\
e^{|x(2)-x(3)|} &\leq 2 \\
f(x) &\leq [e^3 \ e^3 \ e^3]^\top
\end{aligned}$$

This problem corresponds to Example 5 in Boyd et al. (2007) after applying the logarithmic change of variables and adding the upper bound  $z^{ub} = [e^3 \ e^3 \ e^3]^\top$ . We can easily show that this MOP is convex. When applying the transformation function  $h(z) = \log(z)$ , the function  $h \circ f$  becomes a linear function and thus remains convex. As  $h(z) = \log(z)$  is a strictly increasing and concave transformation, we can apply the results obtained in Section 6. Note that although we only use one example to show the effects of transforming the objectives, this transformation can be applied to any geometric program. In this test case, we did not use the transformation for the selection of facets. Furthermore, as the test case is intended to show the benefits of applying transformations and not for comparing different algorithms, we only used the ENHANCED-algorithm.

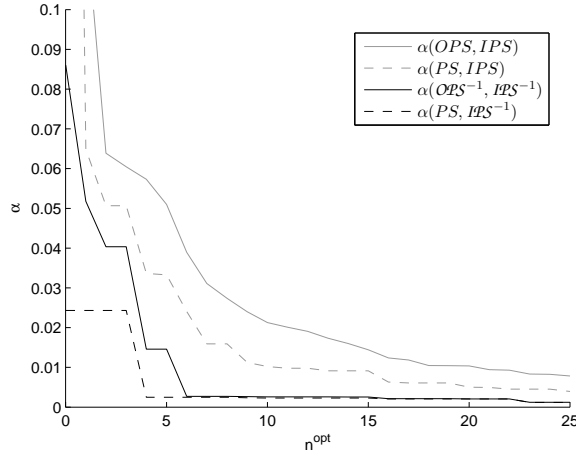


Figure 16:  $\alpha$ -values of test case 4 as function of  $n^{opt}$ .

Figure 16 shows the values of  $\alpha(OPS, IPS)$ ,  $\alpha(P, IPS)$ ,  $\alpha(OPS^{-1}, IPS^{-1})$ , and  $\alpha(P, IPS^{-1})$  for different values of  $n^{opt}$ . Comparing  $\alpha(OPS, IPS)$  with  $\alpha(OPS^{-1}, IPS^{-1})$  and  $\alpha(P, IPS)$  with  $\alpha(P, IPS^{-1})$  shows that both the upper bound and the real accuracy improves significantly by applying the transformation. At  $n^{opt} = 4$ , the real accuracy  $\alpha(P, IPS^{-1})$  is already

equal to 0.0025. After three more optimizations, the value of  $\alpha(OPS^{-1}, IPS^{-1})$  is almost the same as the real accuracy. Without the transformations, more than 25 optimizations are needed to reach the same level of accuracy.

## 9 Conclusions and future research

In this paper, we have introduced several enhancements to improve sandwich algorithms for approximating higher dimensional convex Pareto sets. Firstly, dummy points were introduced to overcome the problem of 'undesirable' normals of  $IPS$ -facets. By adding these dummy points also the set  $IPS^{nId}$  of non- $IPS$ -dominated points can be determined more easily. Secondly, we introduced  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$  which can be used to determine when a set  $IPS^{nId}$  is both an  $\epsilon$ -approximation and  $\epsilon$ -optimal. As both concepts of  $\epsilon$ -approximation and  $\epsilon$ -optimal have a clear interpretation, the measure provides quality guarantees which are easy to understand by a decision maker. Furthermore, we introduced a method to calculate  $\alpha(PS, IPS)$  and  $\alpha(OPS, IPS)$ . This method simplifies the calculation of  $\alpha(OPS, IPS)$  to solving a finite number of LP-problems and thus improves the practical applicability of this error measure. Thirdly, we showed that transformations of the objective functions can be used to improve  $OPS$  and  $IPS$  for certain convex MOPs and to extend the application of sandwich algorithm to certain non-convex MOPs. Also the calculation of the error measure when using these transformations was discussed.

To test the benefit of these enhancements, we constructed the algorithm ENHANCED by enhancing SOLANKI with dummy points and the error measure  $\alpha(OPS, IPS)$ . Three studied test cases showed a significant improvement in efficiency of ENHANCED over the other four tested methods. A fourth test case shows that using suitable transformations can still further improve the efficiency.

A limitation of this comparison is that it only considers sandwich-algorithms. Although non-sandwich algorithms for approximating convex Pareto sets generally cannot provide quality guarantees, they still can provide good approximations of the Pareto set. Therefore, it would be interesting to perform a more extensive comparison between different methods for approximating higher dimensional convex Pareto sets.

Another interesting subject for further research would be to determine if the efficiency could be further improved by using a more interactive approach. In Klamroth and Miettinen (2008), an approach is described where the decision maker can refine his preferences to identify regions of  $PS$  where the approximation should be improved. A similar refinement might also be incorporated into the sandwich approaches by allowing the decision maker to change  $z^{ub}$ . However, more research would be necessary to determine the effects and benefits of such an interactive approach.

## Acknowledgements

The authors wish to thank David Craft for providing the data and problem description of the 2D phantom pancreas IMRT-case. Furthermore, the authors thank Aswin Hoffmann for the many inspiring conversations on the topic of this paper and Dolf Talman for his comments on an earlier version.

## References

- Barber, C.B., D.P. Dobkin, and H. Huhdanpaa (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)* **22(4)**, 469–483.
- Boyd, S., S.J. Kim, L. Vandenbergh, and A. Hassibi (2007). A tutorial on geometric programming. *Optimization and Engineering* **8(1)**, 67–127.
- Branke, J., K. Deb, K. Miettinen, and R. Slowinski (2008). *Multiobjective Optimization - Interactive and Evolutionary Approaches, LNCS 5252, State-of-the-Art Survey*. Springer.
- Carlyle, W.M., J.W. Fowler, E.S. Gel, and B. Kim (2003). Quantitative comparison of approximate solution sets for bi-criteria optimization problems. *Decision Sciences* **34(1)**, 63–82.
- Craft, D.L., T.F. Halabi, H.A. Shih, and T.R. Bortfeld (2006). Approximating convex Pareto surfaces in multiobjective radiotherapy planning. *Medical Physics* **33(9)**, 3399–3407.
- Das, I. and J.E. Dennis (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization* **14(1)**, 63–69.
- Ehrgott, M. (2005). *Multicriteria Optimization*. Springer.
- Ehrgott, M. and M.M. Wiecek (2005). Multiobjective Programming. In J. Figueira, S. Greco, and M. Ehrgott (Eds.), *Multiple Criteria Decision Analysis: State of the Art Surveys*, 667–722. Boston, Dordrecht, London: Springer Verlag.
- Evtushenko, Y.G. and M.A. Potapov (1987). Methods of numerical solution of multicriterion problem. *Soviet mathematics – doklady* **34**, 420–423.
- Hoffmann, A.L., D. den Hertog, A.Y.D. Siem, J.H.A.M. Kaanders, and H. Huizenga (2008). Convex reformulation of biologically-based multi-criteria intensity-modulated radiation therapy optimization including fractionation effects. *Physics in Medicine and Biology* **53(22)**, 6345–6362.
- Hoffmann, A.L., A.Y.D. Siem, D. den Hertog, J.H.A.M. Kaanders, and H. Huizenga (2006). Derivative-free generation and interpolation of convex Pareto optimal IMRT plans. *Physics in Medicine and Biology* **51**, 6349–6369.
- Hwang, C.L. and ASM Masud (1979). *Multiple Objective Decision Making-Methods and Applications: A State of the Art Survey*, Volume 164 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag.
- Jin, Y. and B. Sendhoff (2004). Constructing dynamic optimization test problems using the multi-objective optimization concept. In G. Raidl et al. (Eds.), *Applications of Evolutionary Algorithms*, Volume 3005 of *LNCS*, 525–536. Springer.
- Karasakal, E. and M. Koksalan (2009). Generating a representative subset of the nondominated frontier in multiple criteria decision making. *Operations Research* **57(1)**, 187–199.
- Klamroth, K. and K. Miettinen (2008). Integrating approximation and interactive decision making in multicriteria optimization. *Operations Research* **56(1)**, 222–234.
- Klamroth, K., J. Tind, and M.M. Wiecek (2002). Unbiased approximation in multicriteria optimization. *Mathematical Methods of Operations Research (ZOR)* **56(3)**, 413–437.

- Loridan, P. (1984).  $\varepsilon$ -solutions in vector minimization problems. *Journal of Optimization Theory and Applications* **43**(2), 265–276.
- Marler, R.T. and J.S. Arora (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* **26**(6), 369–395.
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Springer.
- Nguyen, T.A. (2007). Application of optimization methods to controller design for active suspensions. *Mechanics Based Design of Structures and Machines* **35**(3), 291–318.
- Reuter, H. (1990). An approximation method for the efficiency set of multiobjective programming problems. *Optimization* **21**(6), 905–911.
- Rockafellar, R.T. and R.J.B. Wets (1998). *Variational Analysis*. Springer.
- Romeijn, H.E. and J. Dempsey (2008). Intensity modulated radiation therapy treatment plan optimization. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research* **16**(2), 215–243.
- Romeijn, H.E., J.F. Dempsey, and J.G. Li (2004). A unifying framework for multi-criteria fluence map optimization models. *Physics in Medicine and Biology* **49**(10), 1991–2013.
- Ruhe, G. and B. Fruhwirth (1990).  $\varepsilon$ -Optimality for bicriteria programs and its application to minimum cost flows. *Computing* **44**(1), 21–34.
- Ruzika, S. and M.M. Wiecek (2003). A survey of approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications* **126**(3), 473–501.
- Schandl, B., K. Klamroth, and M.M. Wiecek (2002). Norm-based approximation in multicriteria programming. *Computers and Mathematics with Applications* **44**(7), 925–942.
- Shao, L. and M. Ehrgott (2008). Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. *Mathematical Methods of Operations Research* **68**(2), 257–276.
- Siem, A.Y.D., D. den Hertog, and A.L. Hoffmann (2008). The effect of transformations on the approximation of univariate (convex) functions with applications to Pareto curves. *European Journal of Operational Research* **189**(2), 347–362.
- Solanki, R.S., P.A. Appino, and J.L. Cohon (1993). Approximating the noninferior set in multiobjective linear programming problems. *European Journal of Operational Research* **68**(3), 356–373.
- Steuer, R.E. (1986). *Multiple Criteria Optimization: Theory, Computation, and Application*. Wiley, New York.
- White, D.J. (1990). A bibliography on the applications of mathematical programming multiple-objective methods. *Journal of the Operational Research Society* **41**(8), 669–691.
- Zitzler, E., L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* **7**(2), 117–132.